MySQL 8.0 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL 8.0, up through MySQL 8.0.18. For information about changes in a different MySQL series, see the release notes for that series.

For additional MySQL 8.0 documentation, see the MySQL 8.0 Reference Manual, which includes an overview of features added in MySQL 8.0 (What Is New in MySQL 8.0), and discussion of upgrade issues that you may encounter for upgrades from MySQL 5.7 to MySQL 8.0 (Changes in MySQL 8.0).

Updates to these notes occur as new product features are added, so that everybody can follow the development process. If a recent version is listed here that you cannot find on the download page (https://dev.mysql.com/downloads/), the version has not yet been released.

The documentation included in source and binary distributions may not be fully up to date with respect to release note entries because integration of the documentation occurs at release build time. For the most up-to-date release notes, please refer to the online documentation instead.

For legal information, see the Legal Notices.

For help with using MySQL, please visit the MySQL Forums, where you can discuss your issues with other MySQL users.

Document generated on: 2019-06-30 (revision: 18254)

Table of Contents

Preface and Legal Notices ......................................................... 1
Changes in MySQL 8.0.18 (Not yet released, General Availability) ......................................................... 3
Changes in MySQL 8.0.17 (Not yet released, General Availability) ......................................................... 3
Changes in MySQL 8.0.16 (2019-04-25, General Availability) ......................................................... 3
Changes in MySQL 8.0.15 (2019-02-01, General Availability) ......................................................... 28
Changes in MySQL 8.0.14 (2019-01-21, General Availability) ......................................................... 28
Changes in MySQL 8.0.13 (2018-10-22, General Availability) ......................................................... 50
Changes in MySQL 8.0.12 (2018-07-27, General Availability) ......................................................... 74
Changes in MySQL 8.0.11 (2018-04-19, General Availability) ......................................................... 90
Changes in MySQL 8.0.5 - 8.0.10 (Skipped version numbers) ......................................................... 113
Changes in MySQL 8.0.4 (2018-01-23, Release Candidate) ......................................................... 114
Changes in MySQL 8.0.3 (2017-09-21, Release Candidate) ......................................................... 148
Changes in MySQL 8.0.2 (2017-07-17, Development Milestone) ......................................................... 169
Changes in MySQL 8.0.1 (2017-04-10, Development Milestone) ......................................................... 194
Changes in MySQL 8.0.0 (2016-09-12, Development Milestone) ......................................................... 236
Index ................................................................................... 267

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL 8.0, up through MySQL 8.0.18.

Legal Notices

Copyright © 1997, 2019, Oracle and/or its affiliates. All rights reserved.
This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.
Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Changes in MySQL 8.0.18 (Not yet released, General Availability)

Version 8.0.18 has no release notes, or they have not been published because the product version has not been released.

Changes in MySQL 8.0.17 (Not yet released, General Availability)

Version 8.0.17 has no release notes, or they have not been published because the product version has not been released.

Changes in MySQL 8.0.16 (2019-04-25, General Availability)

- Account Management Notes
- C API Notes
- Character Set Support
- Compilation Notes
- Configuration Notes
- Deprecation and Removal Notes
- Installation Notes
- Packaging Notes
- Parser Notes
- Performance Schema Notes
- Plugin Notes
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
- Test Suite Notes
Account Management Notes

• Previously, users who had the **DROP ROLE** privilege could use the **DROP ROLE** statement to drop locked or unlocked accounts. Now, users who have the **DROP ROLE** privilege can use **DROP ROLE** only to drop accounts that are locked (unlocked accounts are presumably user accounts used to log in to the server and not just as roles). Users who have the **CREATE USER** privilege can use **DROP ROLE** to drop accounts that are locked or unlocked. (Bug #28953158, Bug #93263)

• Several changes have been made to MySQL account-management capabilities:

  • MySQL now incorporates the concept of user account categories, with system and regular users distinguished according to whether they have the new **SYSTEM_USER** privilege:

    • System users are users who possess the **SYSTEM_USER** privilege. A system user can perform operations on both system and regular accounts.

    • Regular users are ordinary users who do not possess the **SYSTEM_USER** privilege. A regular user can perform operations on regular accounts, but not system accounts.

  If a user has the appropriate privileges to perform a given operation on regular accounts, **SYSTEM_USER** enables the user to also perform the operation on system accounts. **SYSTEM_USER** does not imply any other privilege, so the ability to perform a given account operation remains predicated on possession of any other required privileges. For example, if a user can grant the **SELECT** and **UPDATE** privileges to regular accounts, then with **SYSTEM_USER** the user can also grant **SELECT** and **UPDATE** to system accounts.

  The distinction between system and regular accounts enables better control over certain account administration issues by protecting accounts that have the **SYSTEM_USER** privilege from accounts that do not have the privilege. For example, the **CREATE USER** privilege enables not only creation of new accounts, but modification and removal of existing accounts. Without the system user concept, a user who has the **CREATE USER** privilege can modify or drop any existing account, including the root account. The concept of system user enables restricting modifications to the root account (itself a system account) so they can be made only by system users. Regular users with the **CREATE USER** privilege can still modify or drop existing accounts, but only regular accounts.

  Other operational implications of the **SYSTEM_USER** privilege:

    • A session that has the **SYSTEM_USER** privilege can be killed only by users who have the **SYSTEM_USER** privilege, in addition to any other required privileges.

    • An account that has the **SYSTEM_USER** privilege can be specified as the **DEFINER** for a stored object only by users who have the **SYSTEM_USER** privilege, in addition to any other required privileges.

    • A role that has the **SYSTEM_USER** privilege cannot be listed in the value of the **mandatory_roles** system variable.

  For more information, see [Account Categories](https://dev.mysql.com/doc/refman/8.0/en/account-categories.html).

• Previously, it was not possible to grant privileges that apply globally except for certain schemas. This is now possible if the new **partial_revokes** system variable is enabled. For example, the following
statements enable an account to select from or insert into any table except those in the `mysql` system schema:

```sql
SET PERSIST partial_revokes = ON;
GRANT SELECT, INSERT ON *.* TO u1;
REVOKE SELECT, INSERT ON mysql.* FROM u1;
```

The server records partial revokes by adding a `Restrictions` attribute to the `User_attributes` column of the `mysql.user` system table. `SHOW GRANTS` includes `REVOKE` statements in its output to indicate partial revokes.

For more information, see `Privilege Restriction Using Partial Revokes`.

- The `GRANT` statement has a new `AS user [WITH ROLE]` clause that specifies additional information about the privilege context to use for statement execution. This syntax is visible at the SQL level, although its primary purpose is to enable uniform replication across all nodes of grantor privilege restrictions imposed by partial revokes, by causing those restrictions to appear in the binary log.

For more information, see `GRANT Syntax`.

**C API Notes**

- The MySQL C API now supports asynchronous functions for nonblocking communication with the MySQL server:
  ```c
  mysql_real_connect_nonblocking()
  mysql_real_query_nonblocking()
  mysql_store_result_nonblocking()
  mysql_next_result_nonblocking()
  mysql_fetch_row_nonblocking()
  mysql_free_result_nonblocking()
  ```
  Each function is the asynchronous counterpart to a synchronous function that has the same name without a `_nonblocking` suffix. The synchronous functions block if reads from or writes to the server connection must wait. The asynchronous functions enable an application to check whether work on the server connection is ready to proceed. If not, the application can perform other work before checking again later. See `C API Asynchronous Interface`.

**Character Set Support**

- MySQL now supports a new Chinese collation, `utf8mb4_zh_0900_as_cs`, for the `utf8mb4` Unicode character set. `utf8mb4_zh_0900_as_cs` is the first Chinese language-specific collation available for Unicode in MySQL. This collation is accent sensitive and case sensitive. Its characteristics are similar to `utf8mb4_0900_as_cs` except that language-specific rules take precedence where applicable. For more information, see `Unicode Character Sets`.

**Compilation Notes**

- `CMake` now causes the build process to link with the `llvm lld` linker for Clang if it is available and not explicitly disabled. To disable use of this linker, specify the `-DUSE_LD_LLD=OFF` option. (Bug #29264211)
MySQL 8.0 Release Notes

• Builds on EL6 and EL7 now try to use the compiler in devtoolset-8 rather than devtoolset-7. (Bug #29198846)

• The minimum version of the Boost library for server builds is now 1.69.0. (Bug #29114233)

• The configuration-time check for Visual Studio 2017 was not specific enough. The check for MySQL compilation now requires at least Visual Studio update 15.8, which is version number 1915. (Bug #28970895)

• MySQL now can be compiled using C++14. The following minimum version requirements apply for compiler support:
  • GCC 5.3 (Linux)
  • Clang 4.0 (FreeBSD)
  • XCode 9 (MacOS)
  • Developer Studio 12.6 (Solaris)
  • Visual Studio 2017 (Windows)

Configuration Notes

• MySQL configuration now requires a minimum CMake version of 3.4.3. This requires the use of cmake3 rather than cmake on some Red Hat and Oracle Linux platforms. (Bug #29246216)

• The WITH_LZMA CMake option was removed. (Bug #29153932, Bug #93755)

• The EXCLUDE_FROM_ALL option is now used in CMake configuration as appropriate so that libraries are built only if they are actually used by any executable. (Bug #29052599)

• The new WITH_JEMALLOC CMake option indicates whether to link with -ljemalloc. If enabled, built-in malloc(), calloc(), realloc(), and free() routines are disabled. The default is OFF. (Bug #29027974)

• The new WITH_LSAN CMake option indicates whether to run LeakSanitizer, without AddressSanitizer. The default is OFF. (Bug #28936574)

• The new WITH_ROUTER CMake option indicates whether to build MySQL Router. The default is ON. (Bug #28759234)

• MySQL Server now supports a --validate-config option that enables the startup configuration to be checked for problems without running the server in normal operational mode. For more information, see Server Configuration Validation.

Deprecation and Removal Notes

• The TempTable storage engine now always uses InnoDB to manage internal temporary tables on disk, and the choice of storage engine employed for this task is no longer user-configurable. The internal_tmp_disk_storage_engine system variable has been removed. (Bug #91377, Bug #28234637)

References: See also: Bug #28081038, Bug #82556, Bug #27408352.
**Installation Notes**

- Previously, after installation of a new version of MySQL, the MySQL server automatically upgraded the data dictionary tables at the next startup, after which the DBA was expected to invoke `mysql_upgrade` manually to upgrade the system tables in the `mysql` schema, as well as objects in other schemas such as the `sys` schema and user schemas.

The server now performs the tasks previously handled by `mysql_upgrade`. After installation of a new MySQL version, the server now automatically performs all necessary upgrade tasks at the next startup and is not dependent on the DBA invoking `mysql_upgrade`. In addition, the server updates the contents of the help tables (something `mysql_upgrade` did not do). A new `--upgrade` server option provides control over how the server performs automatic data dictionary and server upgrade operations. For more information, see Upgrading MySQL.

This change to the upgrade procedure results in some deprecations:

- `mysql_upgrade` is deprecated because it is no longer necessary.
- The `--no-dd-upgrade` server option is deprecated because the `--upgrade` option supersedes it.

`mysql_upgrade` and the `--no-dd-upgrade` option will be removed in a future MySQL version. (Bug #28146052, Bug #28162609, Bug #91205, Bug #29185739, Bug #27740692, Bug #28547424, Bug #91961)

**Packaging Notes**

- Binary packages that include `curl` rather than linking to the system `curl` library now use `curl` 7.64.0. (Bug #29357198)
- The Henry Spencer regex library (`extra/regex`) is no longer used by MySQL 8.0 and is no longer present in source distributions. (Bug #29192306)
- RPM packages now have dependencies on `libtirpc` and `rpcgen` because newer versions of `glibc` do not include Sun RPC. (Bug #28995257)
- The `support-files/magic` file was removed from the MySQL source tree. Most MySQL file formats are covered by operating system file type capabilities. (Bug #18335080, Bug #71898)
- MySQL now provides “Minimal Install” Linux generic binary download packages for MySQL Server and the Test Suite. Minimal install packages exclude debug binaries and are stripped of debug symbols, making them significantly smaller than the regular Linux generic binary packages. Downloads are available at [https://dev.mysql.com/downloads/](https://dev.mysql.com/downloads/).
- The Docker image for MySQL Enterprise Edition is now available for download from My Oracle Support.

**Parser Notes**

- The parser no longer accepts the undocumented and nonstandard `= alias_name` syntax for specifying table aliases. (Bug #29205289)
- The parser now permits several additional nonreserved keywords to be used as labels within stored programs that previously were restricted from such use: `ACCOUNT`, `ALWAYS`, `BACKUP`, `CLOSE`, `FORMAT`, `GROUP_REPLICATION`, `HOST`, `INVISIBLE`, `OPEN`, `OPTIONS`, `OWNER`, `PARSER`, `PORT`, `REMOVE`, `RESTORE`, `ROLE`, `SECONDARY`, `SECONDARY_ENGINE`, `SECONDARY_LOAD`, `SECONDARY_UNLOAD`, `SECURITY`, `SERVER`, `SOCKET`, `SONAME`, `UPGRADE`, `VISIBLE`, `WRAPPER`. (Bug #29033659)
• The parser accepted ODBC escape syntax for outer joins (\{ OJ outer_join \}), but also accepted
identifiers other than OJ. The parser now accepts only OJ.

Note
OJ now is a nonreserved keyword.

(Bug #22320942)

Performance Schema Notes

• MySQL now includes built-in SQL functions that format or retrieve Performance Schema data, and that
may be used as equivalents for existing sys schema stored functions:

  • FORMAT_BYTES(): Converts a byte count to a value with units. Similar to sys.format_bytes().
  • FORMAT_PICO_TIME(): Converts a time in picoseconds to a value with units. Similar to
    sys.format_time().
  • PS_THREAD_ID(): Returns the Performance Schema thread ID for a given thread. Similar to
    sys.ps_thread_id() invoked with a non-NULL argument.
  • PS_CURRENT_THREAD_ID(): Returns the Performance Schema thread ID for the current thread.
    Shortcut for sys.ps_thread_id() invoked with a NULL argument.

The built-in functions can be invoked in any schema and require no qualifier, unlike the sys functions,
which require either a sys. schema qualifier or that sys be the current schema.

The built-in functions supersede the corresponding sys functions, which now are deprecated and will be
removed in a future MySQL version. Applications that use the sys functions should be adjusted to use
the built-in functions instead, keeping in mind some minor differences between the sys functions and the
built-in functions. See Performance Schema Functions.

• The new Performance Schema keyring_keys table exposes metadata for keys in the MySQL Keyring.
  Key metadata includes key IDs, key owners, and backend key IDs. The keyring_keys table does not
  expose any sensitive keyring data such as key contents. See The keyring_keys table.

Plugin Notes

• MySQL now includes a ddl_rewriter plugin that modifies CREATE TABLE statements received by
  the server before it parses and executes them. The plugin removes ENCRYPTION, DATA DIRECTORY,
  and INDEX DIRECTORY clauses, which may be helpful when restoring tables from SQL dump files
  created from databases that are encrypted or that have their tables stored outside the data directory.
  For example, the plugin may enable restoring such dump files into an unencrypted instance or in
  an environment where the paths outside the data directory are not accessible. When installed,
  ddl_rewriter exposes the Performance Schema memory/rewriter/ddl_rewriter instrument for
  tracking plugin memory use. For more information, see The ddl_rewriter Plugin.

Security Notes

• Previously, if the grant tables were corrupted, the MySQL server wrote a message to the error log but
  continued as if the --skip-grant-tables option had been specified. This resulted in the server
  operating in an unexpected state unless --skip-grant-tables had in fact been specified. Now,
  the server stops after writing a message to the error log unless started with --skip-grant-tables.
  (Starting the server with that option enables you to connect to perform diagnostic operations.) (Bug
  #29394501, Bug #94394)
The OpenSSL libraries bundled with MySQL on some platforms (Windows, macOS, and Generic Linux) have been upgraded to version 1.0.2r. On all other platforms, MySQL uses the system installed OpenSSL. Issues fixed in the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html. (Bug #28988091)

Granting roles to anonymous users is no longer supported because such roles caused problematic behavior. (Bug #28910120)

OpenSSL 1.1.1 supports the TLS v1.3 protocol for encrypted connections, and MySQL now supports TLS v1.3 as well, if compiled using OpenSSL 1.1.1 or higher:

- Some TLSv1.3 ciphersuites are enabled by default. The `tls_ciphersuites` system variable enables explicitly specifying which TLSv1.3 ciphersuites the server permits.

- The `--tls-ciphersuites` client option specifies which TLSv1.3 ciphersuites a client permits. This option applies to these programs: `mysql`, `mysqladmin`, `mysqlbinlog`, `mysqlcheck`, `mysql dump`, `mysqldump`, `mysqlimport`, `mysqlpump`, `mysqldump`, `mysqlshow`, `mysqlslap`, `mysqltest`, `mysql_secure_installation`, and `mysql_upgrade`.

- The `mysql_options()` C API function has a new `MYSQL_OPT_TLS_CIPHERSUITES` option that specifies from within the client library which TLSv1.3 ciphersuites a client program permits.

For more information, see Encrypted Connection Protocols and Ciphers.

**Note**

Currently, Group Replication does not support TLSv1.3.

The SSL context the server uses for new connections now is reconfigurable at runtime. This capability may be useful, for example, to avoid restarting a MySQL server that has been running so long that its SSL certificate has expired. Dynamic SSL reconfigurability is based on these changes:

- The system variables that define the SSL context are dynamic and can be changed at runtime: `ssl_ca`, `ssl_capath`, `ssl_cert`, `ssl_cipher`, `ssl_crl`, `ssl_crlpath`, `ssl_key`, `tls_ciphersuites`, `tls_version`.

- The `ALTER INSTANCE` statement supports a `RELOAD TLS` action that reconfigures the SSL context from the current values of the system variables that define the context.

- These status variables reflect the SSL context the server uses for new connections:


  `ALTER INSTANCE RELOAD TLS` updates those status variables from the corresponding system variable values when it reconfigures the SSL context.

For more information, see Server-Side Runtime Configuration for Encrypted Connections.

Thanks to Facebook for submitting code for a similar feature (although the code was not used).

References: See also: Bug #27980097.

In MySQL 8.0, the default authentication plugin was changed from `mysql_native_password` to `caching_sha2_password`. Because `caching_sha2_password` provides a superset of the capabilities of the `sha256_password` authentication plugin, `sha256_password` is now
MySQL 8.0 Release Notes

deprecated and will be removed in a future MySQL version. MySQL accounts that authenticate using `sha256_password` should be migrated to use `caching_sha2_password` instead.

Spatial Data Support

- The `ST_Length()` function now takes an optional second argument enabling the unit for the return value to be specified. Permitted units are those listed in the new `INFORMATION_SCHEMA ST_UNITS_OF_MEASURE` table. See LineString and MultiLineString Property Functions, and The `INFORMATION_SCHEMA ST_UNITS_OF_MEASURE` Table.

SQL Syntax Notes

- **Incompatible Change:** In MySQL 5.7, specifying a `FOREIGN KEY` definition for an InnoDB table without a `CONSTRAINT symbol` clause, or specifying the `CONSTRAINT` keyword without a `symbol`, causes InnoDB to use a generated constraint name. That behavior changed in MySQL 8.0, with InnoDB using the `FOREIGN KEY index_name` value instead of a generated name. Because constraint names must be unique per schema (database), the change caused errors due to foreign key index names that were not unique per schema. To avoid such errors, the new constraint naming behavior has been reverted, and InnoDB once again uses a generated constraint name.

  For consistency with InnoDB, the NDB storage engine now uses a generated constraint name if the `CONSTRAINT symbol` clause is not specified, or the `CONSTRAINT` keyword is specified without a `symbol`. In NDB releases based on MySQL 5.7 and earlier MySQL 8.0 releases, NDB used the `FOREIGN KEY index_name` value.

  The changes described above may introduce incompatibilities for applications that depend on the previous foreign key constraint naming behavior. (Bug #29173134)

  - Previously, MySQL permitted a limited form of `CHECK` constraint syntax, but parsed and ignored it. MySQL now implements the core features of table and column `CHECK` constraints, for all storage engines. Constraints are defined using `CREATE TABLE` and `ALTER TABLE` statements. For more information, see CHECK Constraints. (Bug #11744849, Bug #3464, Bug #3465, Bug #11746042, Bug #22759)

Test Suite Notes

- `mysql-test-run.pl` now supports the `MTR_UNIQUE_IDS_DIR` environment variable, which may be set to specify a unique-IDs directory to be used as the common location for all `chroot` environments by multiple simultaneous `mysql-test-run.pl` instances. This enables those instances to avoid conflicts when reserving port numbers. Thanks to Facebook for the contribution. (Bug #29221085, Bug #93950)

  - The `my_safe_process` program was renamed to `mysqltest_safe_process` and now is installed with other binaries such as `mysqltest` rather than with test suite files. (Bug #29198969)

  - These changes to the `all_persisted_variables` test were implemented:

    - It limits the number of hardcoded values in the test output by putting them into local variables. New patches that are then rebased on top that add new system variables do not need to change as many lines of the original test case, making it easier on the rebase process.

    - It removes entries for bugs that were fixed and modifies the queries to include the system variables that were not tested due to open bugs.

    Thanks to the Facebook team for the contribution. (Bug #29013375, Bug #93478)
MySQL 8.0 Release Notes

X Plugin Notes

• X Plugin previously returned a StmtExecuteOk message to the client after query cleanup had ended and the session had been deactivated. The message is now returned as soon as the result is known and before query cleanup, which gives a noticeable improvement in performance. (Bug #28997370)

• X Plugin logged the system message "X Plugin ready for connections" when user connections were not available because preparation of I/O interfaces had failed. (Bug #28906360)

• Some items in the X Plugin code were not instrumented for the Performance Schema by default. (Bug #28898155)

• X Protocol now supports the COM_RESET_CONNECTION utility command to reset the session state without re-authenticating or closing the connection. (Bug #28732455)

• X Plugin produced compilation warnings when the MySQL Server source code was built using the Clang 8 compiler. (Bug #28732158)

Functionality Added or Changed

• InnoDB: When the amount of memory occupied by the TempTable storage engine exceeds the limit defined by the temptable_max_ram variable, the TempTable storage engine allocates space for internal in-memory temporary tables as memory-mapped temporary files. This behavior is now controlled by the temptable_use_mmap variable, which can be disabled to have the TempTable storage engine use InnoDB on-disk internal temporary tables instead. For more information, see Internal Temporary Table Use in MySQL. (Bug #28944457)

• InnoDB: undo and purge subsystem counters were added for monitoring background activities associated with undo log truncation. For counter names and descriptions, query the INFORMATION_SCHEMA.INNODB_METRICS table.

```
SELECT NAME, SUBSYSTEM, COMMENT FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME LIKE '%truncate%';
```

For information about enabling counters and querying counter data, see InnoDB INFORMATION_SCHEMA Metrics Table. (Bug #28813526)

• InnoDB: The new innodb_spin_wait_pause_multiplier variable provides greater control over the duration of spin-lock polling delays that occur when a thread waits to acquire a mutex or rw-lock. Delays can now be tuned more finely to account for differences in PAUSE instruction duration on different processor architectures. For more information, see Configuring Spin Lock Polling.

• InnoDB: An internal service interface was added to support tracking of modified pages.

• InnoDB: The InnoDB data-at-rest encryption feature now supports encryption of the mysql system tablespace. The mysql system tablespace contains the mysql system database and the MySQL data dictionary tables.

• Some InnoDB memory allocation functions that previously were evaluated at runtime now are evaluated at compile time, resulting in performance improvements. (Bug #29370811, Bug #94380)

• The semi-join optimizations for IN subqueries have been extended to work with EXISTS subqueries as well; these can now be handled with the same semi-join strategies as IN subqueries, including first-match, materialization, duplicate weedout and loose index scan.

In addition, the optimizer decorrelates trivially-correlated equality predicates in the WHERE condition attached to the subquery, so that they can be treated similarly to expressions in IN subqueries. The decorrelation is now also performed for IN subqueries as well as EXISTS subqueries.
All hints and optimizer switches applicable to `IN` subqueries which are transformed into semi-join operations are also applicable to transformed `EXISTS` subqueries. All limitations on such optimization of `IN` subqueries also apply to transformed `EXISTS` subqueries, so that, for example, aggregate `EXISTS` subqueries cannot be transformed.

For more information, see Optimizing `IN` and `EXISTS` Subquery predicates with Semi-Join Transformations. (Bug #28805105, Bug #28857990)

- For consistency with the SQL standard and other RDBMS, table aliases are now supported in single-table as well as multi-table `DELETE` statements. (Bug #27455809)

- The group communication engine for Group Replication (XCom, a Paxos variant) includes a cache for messages (and their metadata) that are exchanged between the group members as a part of the consensus protocol. Among other functions, the message cache is used for recovery by members that return to the group after a period where they were unable to communicate with the other group members.

Previously, the size limit for the message cache was fixed at 1GB of memory, and the maximum number of messages in the cache was also fixed. However, now that the `group_replication_member_expel_timeout` system variable (introduced in MySQL 8.0.13) can be set to allow up to an hour for members to return to the group rather than being expelled, having a fixed 1GB limit on the size of the cache could cause such nodes to be unable to recover the messages they missed on re-establishing communication.

For this reason, from MySQL 8.0.16, XCom's message cache has no fixed limit on the number of messages it can contain, and is only bounded by the limit set for the amount of memory it can use. The cache size limit can be set using the new `group_replication_message_cache_size` system variable, which has a default and minimum setting of 1GB, as used in previous MySQL Server versions. If the cache size limit is reached, XCom removes the oldest entries that have been decided and delivered. The cache size limit can be increased or reduced dynamically at runtime. If you reduce the cache size limit, XCom removes the oldest entries that have been decided and delivered until the current size is below the limit. Group Replication's Group Communication System (GCS) alerts you, by a warning message, when a message that is likely to be needed for recovery by a member that is currently unreachable is removed from the message cache. (Bug #26482507)

- Large messages sent between Group Replication group members can now be split into multiple messages when they exceed a user-defined threshold size. Sending an abnormally large message can result in some group members being reported as failed and expelled from the group. This is because the single thread used by the group communication engine (XCom, a Paxos variant) is occupied processing the message for too long, so some of the group members might report the receiver as failed.

The new system variable `group_replication_communication_max_message_size` can be used to specify the maximum message size for Group Replication communications. Messages greater than this size are automatically split into fragments that are sent separately and reassembled by the recipients. Message delivery for a fragmented message is considered complete when all the fragments of the message have been received and reassembled by all the group members. Fragmentation is applied by default, and can be switched off by specifying a zero value for the system variable.

Because older MySQL Server releases do not support message fragmentation, in order to ensure backward compatibility, Group Replication now has the concept of a communication protocol for the group. The communication protocol version is set to accommodate the oldest MySQL Server version that you want the group to support. A MySQL server at version X can only join and reach `ONLINE` status in a replication group if the group's communication protocol version is less than or equal to X.
You can inspect the communication protocol in use by a group by using the new `group_replication_get_communication_protocol()` UDF, which returns the oldest MySQL Server version that the group supports. Versions from MySQL 5.7.14 allow compression of messages, and versions from MySQL 8.0.16 also allow fragmentation of messages. When a new member joins a replication group, it checks the communication protocol version that is announced by the existing members of the group. If the joining member supports that version, it joins the group and uses the communication protocol that the group has announced, even if the member supports additional communication capabilities. If the joining member does not support the communication protocol version, it is expelled from the group.

If you need to change the communication protocol version of a group so that members at earlier releases can join, use the new `group_replication_set_communication_protocol()` UDF to specify the MySQL Server version of the oldest member that you want to allow. This makes the group fall back to a compatible communication protocol version if possible. If you upgrade all the members of a replication group to a new MySQL Server release, the group's communication protocol version is not automatically upgraded to match. If you no longer need to support members at earlier releases, use the `group_replication_set_communication_protocol()` UDF to set the communication protocol version to the new MySQL Server version to which you have upgraded the members. (Bug #26438884, Bug #23240361, Bug #28474580, Bug #91830, Bug #28642504, Bug #26941977, Bug #29240931)

**For Group Replication, the new system variable `group_replication_autorejoin_tries` lets you specify the number of tries that a member makes to automatically rejoin the group if it is expelled, or if it is unable to contact a majority of the group before the `group_replication_unreachable_majority_timeout` setting is reached. The default setting, 0, means that the member does not try to rejoin, and proceeds to the action specified by the `group_replication_exit_state_action` system variable.**

Activate auto-rejoin if you can tolerate the possibility of stale reads and want to minimize the need for manual intervention, especially where transient network issues fairly often result in the expulsion of members. If you specify a number of tries, when the member's expulsion or unreachable majority timeout is reached, it makes an attempt to rejoin (using the current plugin option values), then continues to make further auto-rejoin attempts up to the specified number of tries. After an unsuccessful auto-rejoin attempt, the member waits 5 minutes before the next try. During the auto-rejoin procedure, the member remains in super read only mode and displays an ERROR state on its view of the replication group. The member can be stopped manually at any time by using a STOP GROUP_REPLICATION statement or shutting down the server. If the specified number of tries is exhausted without the member rejoining or being stopped, the member proceeds to the action specified by the `group_replication_exit_state_action` system variable, which can be either remaining in super read only mode or shutting down. (Bug #25673350, Bug #84784, Bug #28732174)

**WHERE conditions making comparisons between constants and column values in which the constant value is out of range or of the wrong type with respect to the column type are now handled during optimization rather than during execution. For example, given a table `t` with a column `c` whose type is TINYINT UNSIGNED, the condition in the query `SELECT * FROM t WHERE c < 256` can be folded to `SELECT * FROM t WHERE TRUE` because 256 is out of range for a column of this type. Comparisons with NULL columns can also be optimized; if the column `c` is nullable, the same query can be optimized as `SELECT * FROM t WHERE c IS NOT NULL`.**

The comparisons that can be treated in this manner are >, >=, <, <=, =, !=, and <=>. (BETWEEN and IN are not currently supported.) Types for which comparisons can be folded based on range and type include integer, floating-point, and fixed-point numeric types. BIT is not supported by this optimization, nor are columns of date and time types.
For more information, see Constant-Folding Optimization. (Bug #90100, Bug #25484743, Bug #29048682, Bug #27703371)

References: See also: Bug #28172538, Bug #29699347.

• Added an experimental tree format for `EXPLAIN` output, which prints the generated iterator tree, and is intended to help users understand how execution was actually set up. `EXPLAIN FORMAT=TREE` is currently unsupported in production and both its syntax and output are subject to change in subsequent versions of MySQL.

• When binary log and relay log encryption is in use on a MySQL server (`binlog_encryption=ON`), you can now rotate the binary log master key at any time while the server is running by issuing `ALTER INSTANCE ROTATE BINLOG MASTER KEY`. You can do this on a regular basis to comply with your organization's security policy, and also if you suspect that the current or any of the previous binary log master keys might have been compromised.

When you rotate the binary log master key, the new master key is used to encrypt the file passwords for the new binary log and relay log files, and subsequent files until the key is changed again. The file passwords for existing encrypted binary log files and relay log files on the server are also re-encrypted in turn using the new binary log master key, starting with the most recent files. Any unencrypted files are skipped. Finally, all binary log encryption keys that no longer apply to any retained binary log files or relay log files are cleaned up from the keyring.

• Table encryption can now be managed globally by defining and enforcing encryption defaults. The `default_table_encryption` variable defines an encryption default for newly created schemas and general tablespaces. The encryption default for a schema can also be defined using the `DEFAULT ENCRYPTION` clause when creating a schema. By default, a table inherits the encryption of the schema or general tablespace it is created in.

Encryption defaults are enforced by enabling the `table_encryption_privilege_check` variable. The privilege check occurs when creating or altering a schema or general tablespace with an encryption setting that differs from the `default_table_encryption` setting, or when creating or altering a table with an encryption setting that differs from the default schema encryption.

The `TABLE_ENCRYPTION_ADMIN` privilege permits overriding default encryption settings when `table_encryption_privilege_check` is enabled.

For more information, see Defining an Encryption Default for Schemas and General Tablespaces.

• The system variable `group_replication_exit_state_action` specifies how Group Replication behaves when a server instance leaves the group unintentionally, for example after encountering an apllier error, or in the case of a loss of majority, or when another member of the group expels it due to a suspicion timing out. Before the system variable was introduced, the server's response in these situations was to switch MySQL to super read only mode by setting the system variable `super_read_only` to ON. `group_replication_exit_state_action` provided the alternative option for the server to shut itself down, which minimizes the possibility of stale reads and means that servers do not need to be pro-actively monitored for failures. When the system variable was introduced in MySQL 5.7.24 and MySQL 8.0.12, the default was set to `READ_ONLY` in 5.7 for compatibility with the existing behavior, and to `ABORT_SERVER` in 8.0. Following user feedback, the default has now been changed to `READ_ONLY` from MySQL 8.0.16.

• When insertions, deletions, or updates are made to partitioned tables, the binary log now records information about the partition and (if any) the subpartition in which the row event took place. A new row event is created for a modification that takes place in a different partition or subpartition, even if the table involved is the same. So if a transaction involves three partitions or subpartitions, three row events
are generated. For an update event, the partition information is recorded for both the “before” image and the “after” image. The partition information is displayed if you specify the -v or --verbose option when viewing the binary log using mysqlbinlog. Partition information is only recorded when row-based logging is in use (binlog_format=ROW).

Bugs Fixed

- **NDB Cluster**: EXPLAIN of a query executed with table access type eq_ref could also show a condition being pushed down, even when condition pushdown was not supported for the query. Now the access type is checked before NDB checks the condition for a possible pushdown optimization.

  This fix does not affect handling of tables which are part of a pushed join, for which NDB continues to support pushed conditions as before. (Bug #27429615)

  References: See also: Bug #27397802, Bug #27808758, Bug #90301.

- **InnoDB**: Undo tablespaces remained unencrypted after enabling undo tablespace encryption at startup. (Bug #29477795)

- **InnoDB**: Problematic macros introduced with undo tablespace DDL support in MySQL 8.0.14 were revised. (Bug #29324132, Bug #94243)

- **InnoDB**: Static thread local variables defined at the wrong scope were not released at thread exit. (Bug #29305186)

- **InnoDB**: The performance_schema.data_locks LOCK_DATA column only showed the secondary index values of the locked record for a lock placed on a unique secondary index, which was not sufficient to ensure the uniqueness of identified records. The clustered index column values of the locked record are now appended. (Bug #29296645)

- **InnoDB**: An incorrect count of transactions using a rollback segment for recovery of an XA transaction prevented an undo tablespace truncation operation from proceeding and left the purge thread busy checking for the undo tablespace to become empty. (Bug #29273194)

  References: This issue is a regression of: Bug #29273194.

- **InnoDB**: Invalid assertions were raised at startup after a failure to retrieve the space ID of a compressed file-per-table tablespace. The invalid assertion code was removed. (Bug #29221385, Bug #93760)

- **InnoDB**: Optimized InnoDB internal temporary tables did not support in-place UPDATE operations, which caused the number of delete-marked records to increase continuously. The large number of delete-marked records could cause longer than expected query execution times. (Bug #29207450)

- **InnoDB**: The std::sort function in the Variance-Aware Transaction Scheduling (VATS) algorithm was replaced by the std::stable_sort function to preserve the original FIFO order for transactions of equal weight. (Bug #29058967)

- **InnoDB**: Write-ahead did not work as expected due to an incorrectly initialized variable.

  Thanks to Yuhui Wang for the contribution. (Bug #29028838, Bug #93442)

- **InnoDB**: The base column information for a generated column was not stored. (Bug #29021730)

- **InnoDB**: An implicit lock check on secondary indexes needlessly compared columns using collation rules. (Bug #29010725)

- **InnoDB**: Assertion code related to the innodb_flush_method O_DIRECT_NO_FSYNC setting was no longer valid due to a recent modification to that setting. Assertion code was revised. (Bug #29007731)
References: See also: Bug #27309336.

- **InnoDB**: When starting the server with undo log encryption enabled, the master key for newly created undo tablespaces was generated without a server UUID. Undo tablespaces should use the DefaultMasterKey if the server UUID is not yet generated. (Bug #29006275)

- **InnoDB**: Data dictionary code did not check for a returned data dictionary object, which could potentially cause the server to exit due to a null pointer access. (Bug #28977444, Bug #93362)

- **InnoDB**: An undo tablespace file was left behind by a failed `CREATE UNDO TABLESPACE` operation. (Bug #28966457)

- **InnoDB**: A `CREATE UNDO TABLESPACE` statement failed on Windows due to an invalid character in the file name. The failure resulted in a hang condition due to a missing `OS_FILE_ON_ERROR_NO_EXIT` attribute in the call that creates the undo tablespace file. (Bug #28955676)

- **InnoDB**: Modifying the value of the `innodb_undo_log_encrypt` variable was not a blocking operation, which could lead to the modification being reverted by a background thread after the operation appeared to have been completed successfully. (Bug #28952870)

- **InnoDB**: An invalid debug assertion was removed from the `temptable::Handler::primary_key_is_clustered` function. (Bug #28949332)

- **InnoDB**: An `ALTER TABLE ... EXCHANGE PARTITION` operation did not properly update column `table_id` values in the data dictionary. (Bug #28927005)

- **InnoDB**: Memory leaks discovered in the `innochecksum` utility were removed. (Bug #28917614, Bug #93164)

- **InnoDB**: A DDL operation that followed a failed attempt to create an index on a virtual column resulted in an assertion failure. (Bug #28825718)

- **InnoDB**: A performance regression was observed for partial update operations on compressed BLOBs less than or equal to 128KB in size. (Bug #28784301)

- **InnoDB**: Running aggregated queries raised `Valgrind` warnings. (Bug #28711717)

- **InnoDB**: A `CHECK TABLE` operation raised an assertion failure. A pointer to a local call stack variable was not set back to null before a function exit. (Bug #28525110)

- **InnoDB**: DDL log functions were modified to handle `ER_TOO_MANY_CONCURRENT_TRXS` errors. (Bug #28523127, Bug #92071)

- **InnoDB**: The purge thread failed to free LOB data pages. (Bug #28510599)

- **InnoDB**: Some DDL log table transactions were not rolled back prior to DDL log recovery. (Bug #28494969)

- **InnoDB**: A function invoked during `SHOW CREATE TRIGGER` processing that retrieves the table name did not perform the expected lowercase conversion. (Bug #28351038)

- **InnoDB**: The `INFORMATION_SCHEMA.INNODB_FOREIGN_TYPE` column reported incorrect values. (Bug #28315651, Bug #91577)

- **InnoDB**: A Linux AIO handler function failed to check if completed I/O events succeeded. Thanks to Wei Zhao for the contribution. (Bug #27850600, Bug #90402)
• **InnoDB**: An assertion failure was raised in a check that determines if a transaction holds an implicit lock on a secondary index. A transaction that does not change the columns of a secondary index that includes virtual columns could be incorrectly determined to hold an implicit lock. (Bug #27491839)

Thanks to Yan Huang for the patch. (Bug #27373959, Bug #89126)

• **InnoDB**: A function called by a `CREATE TABLE` thread attempted to access a table object after it was freed by a background thread.

• **InnoDB**: Two sessions concurrently executing an `INSERT ... ON DUPLICATE KEY UPDATE` operation generated a deadlock. During partial rollback of a tuple, another session could update it. The fix for this bug reverts fixes for Bug #11758237, Bug #17604730, and Bug #20040791. (Bug #25966845)

• **InnoDB**: When the method used to access a joined table was `const`, InnoDB attempted to unlock the matching row multiple times. (Bug #20939184)

• **InnoDB**: The `INDEX_LENGTH` value in `INFORMATION_SCHEMA.TABLES` was not updated when adding an index. (Bug #19811005)

• **Partitioning**: Some partitioning DDL statements were improperly rejected due to name validation checks which used the wrong table identifier. (Bug #29317007)

• **Partitioning**: While rolling back `ALTER TABLE ... COALESCE PARTITION`, the server sometimes attempted to lock and close partitions which had been dropped as a result of this statement. (Bug #28517446)

• **Partitioning**: An `AUTO_INCREMENT` key added to a partitioned table by an `ALTER TABLE` statement using `ALGORITHM=INPLACE` restarted on each partition. (Bug #92241, Bug #28573894)

• **Replication**: Group Replication did not correctly handle the situation where `START GROUP REPLICATION` was issued when the port specified for the member's Group Replication local address was currently busy. (Bug #29347285)

• **Replication**: If the `WAIT_FOR_EXECUTED_GTID_SET()` function was used with a timeout value including a fractional part (for example, 1.5), an error in the casting logic meant that the timeout was rounded down to the nearest whole second, and to zero for values less than 1 second (for example, 0.1). The casting logic has now been corrected so that the timeout value is applied as originally specified with no rounding. Thanks to Dirkjan Bussink for the contribution. (Bug #29324564, Bug #94247)

• **Replication**: The consistency level `AFTER` for the system variable `group_replication_consistency` did not include the consistency guarantees provided by `BEFORE_ON_PRIMARY_FAILOVER`. These consistency guarantees, which were already implicitly present with the `BEFORE` and `BEFORE_AND_AFTER` consistency levels, are now provided with `AFTER`. (Bug #29315752)

• **Replication**: On Debian-based platforms (such as Ubuntu), if the hostname resolved to 127.0.1.1 - which is the default on these platforms - it was not possible to create a cluster using the default settings. Now, in such situations a proper validation of the instance is performed before creating a cluster and adding instances to it. (Bug #29246110)

• **Replication**: In a blocked group, if you set an invalid value for `group_replication_force_members` and then issued `STOP GROUP_REPLICATION`, the server could stop unexpectedly. (Bug #29119961)

• **Replication**: MySQL Server's behavior when the `mysql.gtid_executed` table cannot be accessed has been refactored to provide appropriate error responses and actions. The MySQL server now ensures that writes are permitted to the `mysql.gtid_executed` table when the server is in read only or super read only mode, so that the binary log file can still be rotated in these modes. If the
mysql.gtid_executed table cannot be accessed for writes, and the binary log file is rotated for any reason other than reaching the maximum file size (max_binlog_size), the current binary log file continues to be used. An error message is returned to the client that requested the rotation, and a warning is logged on the server. If the mysql.gtid_executed table cannot be accessed for writes and max_binlog_size is reached, the server responds according to its binlog_error_action setting. If IGNORE_ERROR is set, an error is logged on the server and binary logging is halted, or if ABORT_SERVER is set, the server shuts down. (Bug #2911514)

**Replication:** When issuing STOP GROUP_REPLICATION while the member was trying to assess whether or not it had lost majority, the server could stop unexpectedly. (Bug #29053128)

**Replication:** When a RESET MASTER TO statement is used to specify the starting index number for binary log files, the maximum number that you can specify has been reduced from the maximum integer value to 2000000000. If the maximum integer value was specified, the server was not able to start up because no further binary log files could be created. The server also previously experienced a segmentation fault in that situation. (Bug #28980788, Bug #28995220)

**Replication:** On overloaded servers there was a possibility that when a member joined the group, the VIEW_CHANGE_LOG_EVENT event which marks that point was not logged in the correct place. This could lead to errors in the data transfer to the newly joining server and data divergence. Now, the VIEW_CHANGE_LOG_EVENT event is logged in the correct place in the binary log. In addition, warnings are logged about the delay in logging the event. (Bug #28971594)

**Replication:** On a replication slave with GTIDs in use and binary logging disabled, an assertion was raised in debug mode when a DDL statement was filtered out by a table filter. (Bug #28965972)

**Replication:** Two issues with the deserialization of statement based replication events in the binary log were corrected. (Bug #28889181, Bug #29028491)

**Replication:** If an applier thread was stopped while it was in the process of opening a table, no error was set, which could result in a segmentation fault or assertion depending on the build type. Error handling is now correctly activated in this situation. (Bug #28864557)

**Replication:** When a host name was specified in the IP address whitelist for Group Replication (group_replication_ip_whitelist), IPv6 addresses were used for name resolution and whitelist comparison when an IPv4 address was also available. An IPv4 address should always be preferred for Group Replication connections. Now, if the host name resolves to an IPv4 address, any IPv6 addresses are not considered for comparison to the whitelist. (Bug #28841543)

**Replication:** With GTIDs in use on the server, the master info log on a replication slave was being synchronized every time the master skipped a transaction using the auto-skip function. The process ends with a dummy heartbeat which is sent to the slave and caused a forced flush to the log, and this could have a large cumulative impact on the write load on the slave. The same issue could occur in a circular replication topology with events that originated from the same server and were therefore ignored, which were also handled by the slave with a forced flush to the log. The slave handling code has now been changed to remove the forced flush for heartbeat events and for ignored events received through circular replication, so that the master info log is only synchronized when appropriate (for example, when a CHANGE MASTER statement is issued, or the binary log is rotated). (Bug #28815555, Bug #85158)

**Replication:** When an ALTER TABLE statement is used with a DEFAULT clause to specify an expression default value for a new column, and the expression default value refers to a nondeterministic function, the statement is unsafe for statement-based replication. Previously, such statements were also evaluated in terms of GTID consistency, which was not the appropriate check as the statements do not impact GTID consistency. Now, these statements are evaluated only for binary logging and are handled depending on the binary logging format in use. When binlog_format is set to STATEMENT, the statement is logged but a warning message is written to the error log. When binlog_format is set
to MIXED or ROW, the statement is not executed and an error message is written to the error log. (Bug #28799939)

- **Replication**: In a replication group configured in single-primary mode (group_replication_single_primary_mode=ON, which is the default), if severe network delays affected the group, it was possible for the primary and the secondaries to reach different decisions on a transaction, which could lead to divergence in the gtid_executed sets on the members. The issue has now been fixed. (Bug #28768550, Bug #28966455, Bug #92690)

- **Replication**: Previously, relay logs could not be rotated manually for the Group Replication group_replication_applier channel using the FLUSH RELAY LOGS statement. Due to this restriction, when encryption was enabled for binary log files and relay log files (binlog_encryption=ON), as available from MySQL 8.0.14, the relay log file in use on that channel could not be rotated immediately if encryption was disabled again. The restriction had a similar impact on binary log master key rotation, as available from MySQL 8.0.16. The restriction has now been removed, and the FLUSH RELAY LOGS statement and corresponding internal requests now operate on the group_replication_applier channel as for any other channel, with the exception that if the request is received while a transaction is being applied, the request is performed after the transaction ends. The requester must wait while the transaction is completed and the rotation takes place. This behavior prevents transactions from being split, which is not permitted for Group Replication. (Bug #28684376)

- **Replication**: When you use the group_replication_force_members system variable to force a new configuration for a group, the group communication engine (XCom) now checks that you have not included any group members that are currently unreachable. If any are found, the reconfiguration is disallowed and an error is returned. (Bug #28678845)

- **Replication**: GRANT statements that were written to the binary log were logged incorrectly in some cases, which could result in a GRANT statement that executed successfully on the master causing an error on the replication slave. (Bug #28643405, Bug #29155451, Bug #93750)

- **Replication**: When a member joined a group on server start, if the join process failed, for example because the server was incompatible with the group, there was a possibility that the offline member could still see another member as being online. Now, in such a situation the information shown in the performance_schema.replication_group_members table is restricted to the local member when it is OFFLINE. (Bug #28533993)

- **Replication**: If a storage engine has the capability to log in STATEMENT format but not in ROW format, when binlog_format is set to STATEMENT, an unsafe SQL statement should be logged and a warning message should be written to the error log. However, such statements were instead not executed and an error message was written to the error log, which is the correct behavior when binlog_format is set to MIXED or ROW. The issue has now been corrected so that unsafe statements are logged with a warning as expected when binlog_format is set to STATEMENT. (Bug #28429993, Bug #73936)

- **Replication**: In the event of the recovery channel failing, unprocessed relay logs were being erased. (Bug #27940732)

- **Replication**: It is possible for a replication group member to go offline briefly, then attempt to rejoin the replication group again before the group has detected its failure and been reconfigured to remove the member. Previously, in this situation, the rejoining member could participate in XCom's consensus protocol if it received and processed messages intended for its pre-crash incarnation. This could cause XCom to deliver different values for the same consensus round, because the rejoining member could make a different decision before and after failure. To prevent this situation, a rejoining member now ignores messages intended for its pre-crash incarnation. (Bug #27383487)

- **Replication**: A replication group member could trigger a local view after being expelled from the group due to a loss of majority. This resulted in a message incorrectly stating that the member had resumed
regular operation after the expulsion. Group Replication now checks before delivering a local view that the member has not been expelled. (Bug #27349236)

- **Replication:** If an invalid value was specified for the `group_replication_communication_debug_options` system variable, the Group Communication System set its corresponding internal variable to `GCS_DEBUG_NONE`, and the server returned the invalid value for a `SHOW VARIABLES` query. The value of the system variable is now checked during server initialization, and if an invalid value was specified, an error message is logged and Group Replication does not start automatically. (Bug #26729404)

- **macOS:** CMake 3.12.4 or higher (which forces `UseModernBuildSystem = NO`) is now required on macOS if building with Xcode rather than Makefiles. (Bug #28893131)

- **Microsoft Windows:** Validity testing for the `named_pipe_full_access_group` system variable did not account for `NULL` values. (Bug #29256690)

- **Microsoft Windows:** When multiple instances of `mysqld` were started with the `--no-monitor` option on the same host for same user, the `SHUTDOWN` command shut down the wrong server process. This fix creates a unique shutdown event name for use with `--no-monitor` by appending the process ID of the process. (Bug #28723675)

- **JSON:** The JSON path parser now propagates errors in the same way as most other components of the MySQL server, returning true on error and false on success. (Bug #28851426)

- **JSON:** Removed an unneeded type lookup in `Json_wrapper::get_datetime()`. (Bug #28851324)

- **The `authentication_ldap_simple` plugin could enforce authentication incorrectly.** (Bug #29637712)

- **RPM package obsoletes were updated to enable successful upgrades from MariaDB to MySQL on EL8.** (Bug #29413354)

- **It was possible for the result of an outer join to contain a non-NULL row where a NULL extended row was expected.** (Bug #29402481)

  References: This issue is a regression of: Bug #27808758.

- **SET PASSWORD FOR ...** could not be executed as a prepared statement. (Bug #29387041, Bug #94416)

- **Builds under Visual Studio could fail while building MySQL Router libraries.** (Bug #29382197)

  References: This issue is a regression of: Bug #29361890.

- **Imported foreign keys did not work if defined before the referenced table.** (Bug #29379078, Bug #94400)

  References: This issue is a regression of: Bug #28493257.

- **When an error was raised while evaluating the `ESCAPE` clause of a `LIKE` expression at resolve time, the error status was not propagated to the caller.** (Bug #29368521)

- **Tablespace files for the `innodb_table_stats_backup` and `innodb_index_stats_backup` metadata backup tables were not removed after an in-place upgrade from MySQL 5.7 to MySQL 8.0.** (Bug #29365552)

  While flattening a subquery, if a predicate which was always false was present, the MySQL Optimizer did not perform any kind of transformation, resulting in a subquery that was not prepared, and which asserted later when it was executed. To resolve this issue, when such a predicate is present in a subquery, the subquery's query expression is now unlinked from the query block. (Bug #29356132)
• Upgrade from MySQL 8.0.11, 8.0.12, or 8.0.13 to MySQL 8.0.14 or 8.0.15 failed if an event, routine, or trigger was defined with the \texttt{ALLOW_INVALID_DATES} SQL mode. The SQL mode identifier in the data dictionary was changed in MySQL 8.0.14, causing a migration failure. (Bug \#29350955)

• RPM builds ignored the \texttt{WITH\_SSL} configuration setting. (Bug \#29347534)

• Length metadata for the \texttt{TO\_SECONDS()} function was not always calculated correctly. (Bug \#29321387)

• Conditions using windowing functions removed due to being always true or false were not always handled correctly. (Bug \#29320484)

• \texttt{SET ROLE} statements could leak memory. (Bug \#29304583)

• Upgrading from MySQL 5.7 to MySQL 8.0 on Windows failed with “Error 197 from SE while migrating tablespaces”. The error was due to an access share violation that occurred when attempting to open a tablespace file. (Bug \#29292860)

  References: This issue is a regression of: Bug \#28642608.

• The \texttt{COMPILATION\_COMMENT\_SERVER} value could be incorrect in RPM packages. (Bug \#29284651)

• \texttt{CREATE TABLE IF NOT EXISTS} failed even if the table already exists if the new table definition had no primary key and the \texttt{sql\_require\_primary\_key} system variable was enabled. (Bug \#29283055, Bug \#94134)

  A delete from a partitioned table from which all partitions had been pruned away was not always handled correctly. (Bug \#29280186)

• The \texttt{CMake} check for the GNU \texttt{gold} linker could fail with Clang. (Bug \#29278244)

• The data dictionary version was incremented for the MySQL 8.0.16 release. (Bug \#29278241)

• An argument of the wrong type to a function used with \texttt{DISTINCT} was not always handled correctly. (Bug \#29277571)

• Length metadata for the \texttt{QUOTE()} function was not always calculated correctly. (Bug \#29276074)

• When evaluating \texttt{GREATEST()} or \texttt{LEAST()}, MySQL checked for correct signedness of the return value before checking it for \texttt{NULL}. (Bug \#29275835)

• When strict SQL mode was not in effect, the values of some string functions returning \texttt{NULL} to indicate a result greater than \texttt{max\_allowed\_packet} were handled inconsistently behaviour, which could result in incorrectly sorted output and possibly other misbehaviors. (Bug \#29272683)

  References: See also: Bug \#97301, Bug \#29133127.

• Logic that checks if upgrade is supported to a particular server version was inverted to check for server versions from which upgrade is not supported. (Bug \#29270297)

• Event creation could store an incorrect repetition interval. (Bug \#29269819, Bug \#94085)

• A \texttt{WHERE} condition containing a view reference or an item created by a transformation was not always handled correctly. (Bug \#29268867, Bug \#29268698, Bug \#28723669, Bug \#29244238)

• \texttt{mysql\_ssl\_rsa\_setup} failed to compile using GCC 9. (Bug \#29245251)

• If \texttt{CMake} finds a \texttt{libtirpc} library that is too old to work with MySQL, it tries to use Sun RPC from \texttt{glibc} instead. (Bug \#29240701)

• The server could fail to write slow queries to the slow query log due to incorrect calculation of query execution time. (Bug \#29232684, Bug \#93963)
MySQL regular expression functions using positions employed internal indexes based on 16-bit chunks, rather than on codepoint positions. (Bug #29231490)

On Windows, the MySQL MSI installer could fail to correctly detect whether the Visual Studio 2015 Redistributable was installed. (Bug #29227209)

SDI JSON files did not include the m_hidden field of Index_impl objects. This made it hard to use the SDI JSON to recreate the CREATE statement for the table because InnoDB adds a number of hidden indexes. The SDI JSON now includes the m_hidden field. This changes SDI format, so the SDI version number was increased to the current server version number. (Bug #29210646, Bug #93914)

The position hint for the last row in a range frame was updated to be one row past the actual last row in the frame. (Bug #29201831)

In a column definition, multiple constraint definitions were not accepted when the first was a CHECK constraint. (Bug #29191994)

Error log information buffered during startup could be buffered too long if the server was performing an upgrade. (Bug #29189532)

For nullable columns, if we find an expression that is always true except when the column is NULL, the expression is folded to column IS NOT NULL. When such an expression was nested, this caused NULL rows to be selected in error. To prevent this from happening, such an expressions when nested is now instead folded to IF(column IS NULL, NULL, TRUE). (Bug #29179604)

PERIOD_ADD() did not handle values greater than 32 bits in length for the period argument correctly on Windows platforms. (Bug #29175262)

When sql_auto_is_null is enabled, a WHERE clause of the form WHERE auto_increment_col IS NULL is rewritten as WHERE auto_increment_col = LAST_INSERT_ID(). This transformation was only performed once per auto-incremented value, which made it difficult to know in advance whether the transformation would be performed. Now the transformation is performed unconditionally whenever sql_auto_is_null is enabled.

In addition, the value returned by LAST_INSERT_ID() is now treated as unsigned, fixing a failure to match an auto-incremented value outside the range of a signed BIGINT. (Bug #29171668)

On Debian and Ubuntu, installation operations in noninteractive mode ignored the root password, resulting in the auth_socket authentication plugin being installed by default. (Bug #29165407)

The harness_plugin_eventlog declaration caused compilation errors in some build environments. (Bug #29160214)

The logic for truncating or extending a decimal constant to the desired number of fractional digits during constant folding was lacking. Extra trailing zeros in the fraction could trigger an attempt to widen the fraction, since the decision to widen was based on the number of non-zero fractional digits, rather than on total number of fractional digits, leading to an assert (in debug builds) in the internal function widen_fraction(). This issue is fixed by identifying where it is possible merely to truncate excess trailing zeroes. In this case, it is not necessary to adjust the comparison operator, and the constant can be replaced by one having fewer trailing zeroes instead. (Bug #29155439)

On Windows, the internal function get_mysql_time_from_str_no_warn() did not always perform proper error checking. (Bug #29155126)

References: See also: Bug #29175262.

Under certain conditions, RENAME TABLE statements that renamed the same table multiple times could raise an assertion or cause a server exit. (Bug #29140407)
• For debug builds, starting the server with --event-scheduler=DISABLED could result in an assertion being raised for certain events. (Bug #29140298, Bug #93719)

• In debug builds, when strict SQL mode was not in effect, the CONCAT() and CONCAT_WS() functions raised an assertion if the result was longer than max_allowed_packet. (Bug #29133127)

• An out-of-range fractional part could produce incorrect timestamps in SET SESSION timestamp statements. (Bug #29120569, Bug #93600)

• The mysql_service_component_sys_variable service could access component system variables but not server or plugin system variables. (Bug #29113463)

• ALTER TABLE ... CONVERT TO CHARACTER SET could produce a memory-access error. (Bug #29058369, Bug #93603)

• The mysql.tablespaces.name column limit was 259 bytes, which was less than required for permitted identifier lengths. The column limit was raised to 268 bytes. (Bug #29053560, Bug #93587)

• For debug builds, for spatial computations that raised an assertion, partition-handling code ignored the error, resulting in a server exit. (Bug #29047811)

• A zero length LOB that was stored externally caused an assertion failure. (Bug #29047795)

• Handling of the COMPILE_DEFINITIONS and COMPILER_FLAGS CMake options was adjusted to avoid cross-compiling failures. (Bug #29041100)

• The server could try to read the first diagnostics area message even when the max_error_count system variable was set to zero, resulting in a memory-access error. (Bug #29031684)

• When comparing DATE values with constant strings, MySQL first tries to convert the string to a DATE and then to perform the comparison. When the conversion failed, MySQL executed the comparison treating the DATE as a string, which could lead to unpredictable behavior. Now in such cases, if the conversion of the string to a DATE fails, the comparison fails with ER_WRONG_VALUE. (Bug #29025656)

References: See also: Bug #95466, Bug #29812087.

• With the --users option, mysqldump wrote CREATE USER and GRANT statements to the output, but too late to apply to the other objects created by the dump. Consequently, restoring the dump file created the user accounts too late to apply to other objects created by the file. mysqldump now writes user accounts to the dump file before other objects. (Bug #29023216)

• For syntax errors at the position of the WITH keyword, the parser error message identified the problem at the incorrect location. (Bug #29022263)

• MySQL builds using recent versions of the International Components for Unicode (ICU) now return ER_REGEXP_ILLEGAL_ARGUMENT rather than the generic error ER_REGEXP_ERROR for malformed regular expressions. (Bug #29016798)

• On Windows, the LDAP library could not be found if the file system was case sensitive. (Bug #29016220)

• Values selected from the TABLE_COMMENT column of the INFORMATION_SCHEMA.TABLES table could be truncated. (Bug #29014272, Bug #93451)

• Data layer memory leaks related to unfreed schemas were discovered in ASAN and Valgrind builds. (Bug #29008688)

• A GROUP BY query with ROLLUP incorrectly raised ER_WRONG_FIELD_WITH_GROUP when used with an ORDER BY clause that contained an expression.
The same issue also led to incorrect ordering of the result from a \texttt{GROUP BY} query with \texttt{ROLLUP} when ordering on a function. (Bug \#29006668, Bug \#29054096)

- The parser accepted multiple \texttt{COLLATE} clauses in generated column definitions. It now accepts a single \texttt{COLLATE} clause. (Bug \#28997518)

- \texttt{INSERT ... ON DUPLICATE KEY UPDATE} could perform incorrect updates. (Bug \#28995498, Bug \#93410)

References: This issue is a regression of: Bug \#26188578.

- A damaged mysql.user table could cause a server exit. (Bug \#28986737)

- Reading rows from the Performance Schema \texttt{data_locks} or \texttt{data_lock_waits} table could return an unbounded number of rows during a scan, resulting in a server exit due to excessive memory allocation. This situation now produces an error. (Bug \#28977428, Bug \#87748)

- Calculating the cost of materialization for a \texttt{LATERAL} join on an empty table led to an assertion. Now in such cases, the query cost is reported as zero. (Bug \#28976533)

- For debug builds, an invalid utf8 character in the comment string for a column type in \texttt{CREATE} or \texttt{ALTER} DDL statements raised an assertion. (Bug \#28972424, Bug \#93321)

- The \texttt{CMake} check for \texttt{tirpc} headers now falls back to using \texttt{pkgconfig}, to enable finding the headers on more Linux platforms. This requires that \texttt{pkgconfig} be installed. (Bug \#28970313, Bug \#93341, Bug \#28997093)

- \texttt{mysqld} could undergo an unplanned shutdown when a component in the \texttt{WHERE} clause of a subquery was non-deterministic, in this particular case due to referencing a user variable that was also set in the same statement. This issue is fixed by ignoring predicates having non-deterministic components.

In addition, since non-deterministic subqueries as a general rule should be evaluated per outer row, these should not be targets for materialization strategy. This is prevented by not decorrelating the non-deterministic predicates. When possible, the same strategy selection for non-semijoin and non-deterministic subqueries is also now enforced. (Bug \#28970261)

- \texttt{ALTER TABLE} statements to change a table storage engine could hang when user-level locks and explicit table locks had been acquired earlier. (Bug \#28966941)

- \texttt{COUNT()} with \texttt{LIMIT ... OFFSET} returned a different result from that obtained with other aggregate functions used with this clause. (Bug \#28961843)

- Extracting the value of the \texttt{MYSQL_HOME} environment variable could change the value of the variable stored following \texttt{MYSQL_HOME} in the environment. (Bug \#28960613)

- Opening and closing a connection multiple times in a client program could cause the client to exit due to improper character set reinitialization. (Bug \#28956360, Bug \#93276)

- \texttt{Item_subselect::walk_body()} now walks \texttt{FROM} clauses. (Bug \#28955358)

- An issue was uncovered in query plan for a query with the following two \texttt{IN} subqueries:
  - A subquery with a derived table containing an outer reference to the topmost query
  - A subquery with a derived table, which was not outer-correlated, was semi-joined, and was not lateral.

For the second subquery: calculation of the map of dependencies neglected to exclude irrelevant lateral tables, and wrongly included dependencies of the derived table in the first subquery. (Bug \#28954838)
• An error generated while evaluating a constant expression in certain GROUP BY queries was not checked for in a timely fashion. (Bug #28949452)

• When comparing a DATE or DATETIME value with a string MySQL first tries to convert the string to the same type (DATE or DATETIME) and then to compare the two as values of that type. If this conversion fails, MySQL raises an Incorrect date value warning and then falls back to comparing the values as strings, which is the expected behavior. In some cases, even though the converted string did not reflect a valid DATE or DATETIME value, no warning was issued so it was compared with the DATE or DATETIME value as a value of that type. Now in such cases, the converted string is always checked for validity as a DATE or DATETIME before being compared, unless the ALLOW_INVALID_DATES server SQL mode has been set explicitly. (Bug #28940878)

• Long client host names could cause unexpected server behavior. (Bug #28936359)

• Substitution of a large number of parameters in a single prepared statement could lead to excessive and unnecessary reallocation of memory. This is fixed by having String::replace() use exponential buffer growth as String::append() does. (Bug #28929977)

• The data dictionary cache could become out of sync with data dictionary contents due to failure to check the result of a transaction-related operation. (Bug #28923782, Bug #93196)

• Determination of the number of online CPUs available to the mysqld process is now more accurate. Thanks to Daniel Black for the contribution. (Bug #28907677, Bug #93144)

• The INFORMATION_SCHEMA.TABLES TABLE_COMMENT column reported the wrong error message due to an invalid view. (Bug #28901919)

• After upgrading, the INFORMATION_SCHEMA.TABLES view reported “View 'view_name' references invalid table(s) or column(s) or function(s) or definer/invoker of view lack rights to use them” for several sys schema views. The upgrade process did not update the referencing view state. (Bug #28901821)

• Creating histograms for large tables (millions of rows) having many distinct column values took an excessive amount of time.

  The fix make this process marginally slower for data sets with few distinct values and consumes slightly more memory than previously, but for large data sets with many distinct values it is significantly faster. (Bug #2888936)

• Upgrade from MySQL 5.7 to MySQL 8.0 failed for instances with a an InnoDB 4K page size due a key length error that should be suppressed during upgrade. (Bug #28884503)

• An incorrectly configured keyring plugin could cause a server exit. (Bug #28876033)

• Two issues were found in cases involving use of the COUNT() function:
  • When the argument passed to COUNT() was a nullable expression, it was possible for the function to return an incorrect result.
  • A query using COUNT() together with an EXISTS subquery returned an incorrect result.
    (Bug #28857990, Bug #29240516)

• Commands for invoking ccache during MySQL builds were generated incorrectly. (Bug #28841612, Bug #92927)

• Previously, for INSERT statements that listed insert column names multiple times, only the first duplicated name was reported. Now, all duplicated names are reported. (Bug #28836669)
• When using `ROLLUP`, the `GROUP BY` may yield `NULL` in some rows. Expressions referencing any of these rows could lose the `NULL`, producing an incorrect result. (Bug #28836345)

• After a failed attempt to open a table, attempts to discover the table from the storage engine did not check for an error return, which could cause the server to hang if an error occurred. (Bug #28828450)

• The server could exit when trying to drop a user who had been granted a particular role. (Bug #28817441)

• MySQL Router compilation failed if MySQL was configured with `-DWITH_MYSQLX=0`. (Bug #28811356)

• A function called during data dictionary upgrade temporarily modified the `avoid_temporal_upgrade` parameter so that a check for old temporal types is always performed. Because multiple threads could call the function concurrently, a check was added to ensure that the parameter value could be safely modified during data dictionary upgrade. (Bug #28805429)

• An `EXISTS` subquery is converted to a semi-join operation. The optimizer chooses a materialization lookup strategy for this semi-join, but because the subquery is not correlated with the outer query block, there were no keys to use for the lookup, which caused the parent query to fail. To solve this issue, we use two equal constant items as keys, to ensure that the materialized query gets the constant as a key (and so that the materialized table consists of at most one row). (Bug #28805105)

• The `Last_query_cost` status variable now shows the cost of queries which contain multiple blocks such as subqueries or unions. (Previously, this variable was set only for simple queries consisting of a single query block.) (Bug #28786951)

• The message displayed by `mysqld` when a fatal signal occurred has been simplified and made more informative. It also does not display the calculation of memory usage, which had become out of date and inaccurate. (Bug #28773322, Bug #92731)

• The `skip_name_resolve` system variable could be persisted using `SET PERSIST_ONLY` to enable it, but not to disable it. (Bug #28749668)

• Removed obsolete and unnecessary condition pushdown handling in iterators except where required by single table update and delete queries. (Bug #28745859)

• `IS NOT NULL` predicates are added as part of early filtering of `NULL` for ref access performed by the Optimizer. For queries having a star-join topology, redundant duplicates of these predicates were added, which made evaluation of affected `WHERE` conditions less efficient. (Bug #28727717)

• `CMake` could generate an incorrect order of system includes when MySQL was configured to use the bundled version of a system library. (Bug #28727631, Bug #92615)

• Some numeric operations involving double to long long conversions could return different results on Windows and Linux. (Bug #28706832)

• An attempt to access a null pointer could occur during prepared statement execution. (Bug #28692136)

• `FROM_UNIXTIME()` returned an out-of-range value if passed an argument that, when rounded up, exceeded the epoch value. Now it returns `NULL` as for other out-of-range values. (Bug #28671811, Bug #92501)

• A replicated DDL operation could result in a 'duplicate entry on primary key' error during recovery on servers started with the `--slave-parallel-workers` option. (Bug #28670843)

• Empty host names in accounts could cause the server to misbehave. (Bug #28653104)

• The MySQL 5.7 to MySQL 8.0 upgrade process did not check for the existence of tablespace files, which could result in an inconsistent data dictionary. (Bug #28642608)
• Type casting of `ENUM` behaved differently in subqueries than not in subqueries. (Bug #28547906, Bug #92173)

• Statements that caused truncation of floating-point values could be executed as prepared statements even when the server SQL mode included `STRICT_TRANS_TABLES`. In addition, if the binary logging mode was `MIXED`, such statements were propagated to the slave, where they caused errors. (Bug #28546855)

• Reducing the value of `thread_cache_size` at runtime did not reduce the size of the thread cache. (Bug #28508923, Bug #92024)

• Some error messages still referred to the `PASSWORD()` function, which has been removed. (Bug #28498714)

• The `Aborted_connects` status variable was not incremented for unsuccessful connection attempts, if connections were managed by the `thread_pool` plugin. (Bug #28490126)

• `mysqladmin shutdown` did not wait for `mysqld` to shut down. (Bug #28466137, Bug #91803)

References: This issue is a regression of: Bug #25364806.

• Specifying `CURRENT_USER` as the user in `GRANT` statements could fail. (Bug #28454014)

• Repeated invocations of stored procedures which executed queries undergoing short-circuit evaluation were not always handled correctly. (Bug #28379655)

• Keyring migration should require only read access to the source keyring, but failed unless the user had write access. (Bug #28339014)

• If a user performing a keyring migration did not have write access to the keyring file, the migration failed but reported success in its final error log message. (Bug #28330922)

• During `FLUSH STATUS` execution, the Performance Schema unnecessarily aggregated session status to global status, causing double counts for some status variables. (Bug #28291258, Bug #91541)

• `mysql_secure_installation` no longer attempts to read a password from the `.mysql_secret` file. This was was created by `mysql_install_db`, a program that has been removed. (Bug #28235716, Bug #91270)

• Some status variable values could temporarily increase before returning to their original value. (Bug #27839644, Bug #90351)

• Client programs did not exit if `--ssl-fips-mode` was given but FIPS mode could not be set to the specified option value. (Bug #27809371)

• Executing `ALTER INSTANCE ROTATE INNODB MASTER KEY` and migrating keys from the `keyring_file` plugin to the `keyring_encrypted_file` plugin could make encrypted tables unusable. (Bug #27760952)

• When upgrading from MySQL Community to Commercial, the `root` password prompt was shown even when the data directory existed. (Bug #27741998)

• In the client/server protocol, malformed packets for prepared statements could go undetected and cause an assertion to be raised. (Bug #27627731)

• Changes were made in session connect and disconnect handling for threads in order to speed up query throughput when using the X Plugin. Previously, a session (`Srv_session`) was attached to and detached from the current thread with every command; now, this thread switch takes place only when
disconnecting the current session, or when the current thread and the thread being attached to belong to different plugins. (Bug #27463277)

- The **PERSIST** and **PERSIST_ONLY** keywords were reserved by mistake. They are now nonreserved keywords. (Bug #25220656)

- Installing and uninstalling a plugin concurrently with client connection activity could cause a server exit. (Bug #22980441)

- The **LOAD_FILE()** function could fail for files for which **stat()** should be considered only advisory, such as files under `/proc`. (Bug #18394503, Bug #72027)

- A query using an **ORDER BY** clause failed silently when the table being queried had an implicit full-text index and the sort buffer was of insufficient size to contain the sorted keys. (Bug #93241, Bug #28940361)

- A UDF returning a string value now sets an explicit return type. Depending on the arguments passed to the function, this is one of **VARCHAR**, **MEDIUMBLOB**, or **LONGBLOB**. (Bug #92890, Bug #28828169)

- Made a comparison in the internal method **Item_result::item_cmp_type()** more efficient. Our thanks to Daniel Black for the contribution. (Bug #92784, Bug #28796107)

- A windowing function employed in an arithmetic expression produced an incorrect result when the query containing it used **DISTINCT**. (Bug #92503, Bug #28672483)

- Some queries involving complex joins leaked file handles. (Bug #90902, Bug #28039829)

- Fedora packaging now supports Fedora 30.

- Ubuntu 14.04 and SLES 11 are EOL, and no longer supported.

**Changes in MySQL 8.0.15 (2019-02-01, General Availability)**

**Bugs Fixed**

- **InnoDB**: After a checkpoint operation persisted modifications to data dictionary metadata, there was potential for new metadata changes to be lost under certain circumstances. (Bug #29120297)

- Group Replication was unable to function in the 8.0.14 release of MySQL Server if IPv6 support was disabled at the operating system level, even if the replication group did not use any IPv6 addresses. The issue is fixed by this release of MySQL Server, 8.0.15. (Bug #29249542, Bug #94004)

- If MySQL was running on the host system and within Docker, it was not possible to update or remove MySQL on the host system. (Bug #28244773, Bug #91405)

**Changes in MySQL 8.0.14 (2019-01-21, General Availability)**

- Account Management Notes
- Audit Log Notes
- Compilation Notes
- Component Notes
- Configuration Notes
- Deprecation and Removal Notes
- Function and Operator Notes
Account Management Notes

Previously, each MySQL user account was permitted to have a single password. MySQL now permits an account to have dual passwords, designated as primary and secondary passwords. This capability enables phased password changes to be performed seamlessly in complex multiple-server systems, without downtime. To support dual-password capability, the `ALTER USER` and `SET PASSWORD` statements now have a `RETAIN CURRENT PASSWORD` clause that saves the current password as the secondary password when you assign an account a new primary password. `ALTER USER` also has a `DISCARD OLD PASSWORD` clause to discard a secondary password that is no longer needed. See Password Management.

Important

The implementation of dual-password capability involves a change to the structure of the `mysql.user` system table. If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this system database change. Until this is done, password changes are not possible.

Audit Log Notes

The audit API now enables applications to add their own message events to the audit log using the new `audit_api_message_emit` component, which includes an `audit_api_message_emit_udf()` user-defined function. See The Audit Message Component.

Compilation Notes

The minimum version of the Boost library for server builds is now 1.68.0. (Bug #28478497)

Component Notes

A new `host_application_signal` component service is available to enable server components to deliver signals to the host application. For example, the service enables replication components to send a shutdown signal to the server.
Configuration Notes

- Previously, the `COMPILATION_COMMENT` CMake option was used by the server (for example, to set the `version_comment` system variable) and by other programs. However, when the value contained the word "server", it was inappropriate for use by other programs. The server now uses the new `COMPILATION_COMMENT_SERVER` option. Other programs continue to use `COMPILATION_COMMENT`. (Bug #28888510)

- The content of the `.gitignore` file has been cleaned up. Much of this file was inherited from its `.bzrignore` predecessor and was not relevant. One implication of this cleanup is that in-source builds are disallowed. (Bug #28341794, Bug #91626)

- MySQL Server now permits a TCP/IP port to be configured specifically for administrative connections. This provides an alternative to the single administrative connection that is permitted on the network interfaces used for ordinary connections even when `max_connections` connections are already established. The administrative network interface has these characteristics:

  - The interface is available only if the `admin_address` system variable is set at startup to indicate the IP address for the administrative interface.
  - The `admin_port` system variable specifies the interface TCP/IP port number (default 33062).
  - There is no limit on the number of administrative connections.
  - Connections are permitted only by users who have the `SERVICE_CONNECTION_ADMIN` privilege.

  The `create_admin_listener_thread` enables DBAs to choose whether the administrative interface is implemented using the listener thread used for ordinary connections (this is the default), or has its own separate thread.

  Thanks to Facebook for suggesting the idea (and for contributing code, although it was not used). (Bug #27847672, Bug #90395)

Deprecation and Removal Notes

- The deprecated `resolveip` and `resolve_stack_dump` utilities have been removed and are no longer included in MySQL distributions. `nslookup`, `host`, or `dig` can be used instead of `resolveip`. Stack traces from official MySQL builds are always symbolized, so there is no need to use `resolve_stack_dump`.

Function and Operator Notes

- The `JSON_ARRAYAGG()` and `JSON_OBJECTAGG()` aggregate functions now can be used as window functions, if an `OVER` clause is present. See Aggregate (GROUP BY) Function Descriptions.

Logging Notes

- A new system variable, `log_slow_extra`, if enabled, causes the server to write additional fields to slow query log lines that provide information about slow statements. In addition, `SET` lines written to the log to indicate statement timestamps now use the time from the beginning of statement execution, rather than the time at the end of execution. See The Slow Query Log. Thanks to Facebook for the contribution on which this feature is based. (Bug #27535580, Bug #89637)

- Binary log files and relay log files can now be encrypted, helping to protect these files and the potentially sensitive data contained in them from being misused by outside attackers, and also from unauthorized viewing by users of the operating system where they are stored.
You enable encryption on a MySQL server by setting the new `binlog_encryption` system variable to `ON`. `OFF` is the default. The system variable sets encryption on for binary log files and relay log files. When you first start the server with encryption enabled, a new binary log encryption key is generated before the binary log and relay logs are initialized. This key is used to encrypt a file password for each binary log file (if the server has binary logging enabled) and relay log file (if the server has replication channels), and further keys generated from the file passwords are used to encrypt the data in the files.

If you activate encryption while the server is running, a new binary log encryption key is generated at that time, and the binary log files and relay log files are rotated so that the new and subsequent files are encrypted. If you deactivate encryption by changing the `binlog_encryption` system variable to `OFF`, the binary log file and relay log files are rotated immediately and all subsequent logging is unencrypted. Previously encrypted files are not automatically decrypted, but the server is still able to read them. (The `SHOW BINARY LOGS` statement now shows whether each binary log file is encrypted or unencrypted.) `SUPER` privileges or the new `BINLOG_ENCRYPTION_ADMIN` privilege are required to activate or deactivate encryption while the server is running.

The encryption algorithm used for the files, the AES (Advanced Encryption Standard) cipher algorithm, is built in to MySQL Server and cannot be configured. The binary log encryption keys used to encrypt the file passwords for the log files are 256-bit keys that are generated specifically for each MySQL server instance using MySQL Server's built-in keyring service. The binary log encryption key that is currently in use on the server is called the binary log master key.

The new `binlog_rotate_encryption_master_key_at_startup` system variable controls whether the binary log master key is automatically rotated when the server is restarted. If this system variable is set to `ON`, a new binary log encryption key is generated and used as the new binary log master key whenever the server is restarted. If it is set to `OFF`, which is the default, the existing binary log master key is used again after the restart.

Note that when encryption is active for a MySQL server instance, only the data at rest that is written to the binary log files and relay log files is encrypted. The data in motion in the replication event stream, which is sent to MySQL clients including `mysqlbinlog`, is always in unencrypted format, so it must be protected in transit by the use of connection encryption. The data in use that is held in the binary log transaction and statement caches during a transaction, and any data that exceeds the space available in those caches and is therefore stored in a temporary file on disk, is also in unencrypted format. The temporary files and caches are deleted when the thread that handles the transaction ends.

- Server logging behavior has changed with respect to error log messages generated prior to processing startup options that specify logging configuration. Previously, the server generated messages with the default timestamp, format, and verbosity level; buffered them; and then flushed them after the error log configuration became known. Because these early messages used the default logging configuration, they could differ from what is specified by the startup options.

Now, the server buffers log events rather than formatted log messages. This enables it to retroactively apply configuration settings to those events after the settings are known, with the result that flushed messages use the configured settings, not the defaults. For more information, see Error Log Message Format.

**Optimizer Notes**

- Previously, derived tables and common table expressions could not contain outer references. Outer references are now permitted.
Packaging Notes

- Ubuntu 18.10 and Fedora 29 install OpenSSL 1.1.1 by default, but OpenSSL 1.1.1 is not fully supported by MySQL. To install MySQL, the OpenSSL 1.0.2 compatibility package must be installed. (Bug #28981868)

Performance Schema Notes

- The Performance Schema statement event tables (events_statements_current, events_statements_history, and events_statements_history_long) now have a STATEMENT_ID column that indicates the query ID maintained by the server at the SQL level. Column values are unique for the server instance because they are generated using a global counter that is incremented atomically.

Pluggable Authentication

- If the LDAP port number is configured as 636 or 3269, the plugin now uses LDAPS (LDAP over SSL) instead of LDAP. The port number is settable using the authentication_ldap_sasl_server_port or authentication_ldap_simple_server_port system variable. (LDAPS differs from startTLS.) (Bug #28743563)

- Previously, for LDAP authentication with proxying, LDAP authentication plugins used the first group name returned by the LDAP server as the MySQL proxy user account name. The authentication string for a MySQL account now can specify a list of groups to match, in preference order, and can optionally map the matching group name to a specified MySQL proxy user name. See LDAP Pluggable Authentication.

Security Notes

- The OpenSSL libraries bundled with MySQL on some platforms (Windows, macOS, and Generic Linux) have been upgraded to version 1.0.2q. On all other platforms, MySQL uses the system installed OpenSSL. Issues fixed in the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html. (Bug #28988091)

- To affect subsequent server restarts, the SET PERSIST and SET PERSIST_ONLY statements enable system variables to be persisted to the mysqld-auto.cnf option file in the data directory. However, some system variables cannot be persisted (for example, because they involve sensitive data). Consequently, they cannot be set at runtime from within a session conducted by a remote administrator, and thus require the administrator to log into the server host and directly modify a my.cnf option file.

MySQL now permits users to perform runtime administration of many previously nonpersistible system variables, so that they can be persisted under certain restrictive conditions. To enable this capability, designate an SSL certificate X.509 Subject value that signifies the ability to persist these restricted system variables, and set the new persist_only_admin_x509_subject system variable to that Subject value. Users who connect to the server using an encrypted connection and supply an SSL certificate with the designated Subject value then can use SET PERSISTONLY to persist persist-restricted system variables. For more information, see Nonpersistible and Persist-Restricted System Variables.

- For most system variables, setting the session value requires no special privileges and can be done by any user to affect the current session. For some system variables, setting the session value can have effects outside the current session and thus is a restricted operation that can be done only by users who have a special privilege. Previously, either SYSTEM_VARIABLES_ADMIN or SUPER qualified as such a privilege, but both privileges also permit operations other than setting session variables. The new
SESSION_VARIABLES_ADMIN privilege makes it possible to grant users only the ability to set restricted session variables without also enabling other operations.

Any operation permitted by SESSION_VARIABLES_ADMIN is also permitted by SYSTEM_VARIABLES_ADMIN or SUPER, so any user who already has one of the latter privileges effectively has SESSION_VARIABLES_ADMIN by implication and need not be granted SESSION_VARIABLES_ADMIN explicitly. However, if a user has been granted SYSTEM_VARIABLES_ADMIN or SUPER only for the purpose of enabling that user to modify restricted session system variables, an administrator can reduce the user’s privilege footprint by revoking SYSTEM_VARIABLES_ADMIN and SUPER, and granting SESSION_VARIABLES_ADMIN instead. For instructions, see System Variable Privileges.

These previously restricted session variables required SYSTEM_VARIABLES_ADMIN or SUPER but now can also be set with SESSION_VARIABLES_ADMIN:

```
binlog_format
binlog_row_image
binlog_row_value_options
binlog_rows_query_log_events
debug
debug_sync
default_collation_for_utf8mb4
explicit_defaults_for_timestamp
gtid_next
histogram_generation_max_mem_size
original_commit_timestamp
sql_log_bin
sql_log_off
sql_require_primary_key
```

These previously unrestricted session variables now are restricted and setting them requires at least SESSION_VARIABLES_ADMIN (they can also be set by users who have SYSTEM_VARIABLES_ADMIN or SUPER):

```
auto_increment_increment
auto_increment_offset
binlog_direct_non_transactional_updates
bulk_insert_buffer_size
character_set_filesystem
character_set_database
collation_database
pseudo_slave_mode
pseudo_thread_id
rbr_exec_mode
transaction_write_set_extraction
```

Spatial Data Support

- The ST_Distance() function now takes an optional third argument enabling the unit for the return value to be specified. Permitted units are those listed in the new INFORMATION_SCHEMA ST_UNITS_OF_MEASURE table. See Spatial Relation Functions That Use Object Shapes, and The INFORMATION_SCHEMA ST_UNITS_OF_MEASURE Table.

SQL Syntax Notes

- A derived table now may be preceded by the LATERAL keyword to specify that it is permitted to refer to (depend on) columns of preceding tables in the same FROM clause. A derived table specified with LATERAL can occur only in a FROM clause, either in a list of tables separated with commas or in a join
specification (JOIN, INNER JOIN, CROSS JOIN, LEFT [OUTER] JOIN, or RIGHT [OUTER] JOIN). Lateral derived tables make possible certain SQL operations that cannot be done with nonlateral derived tables or that require less-efficient workarounds. See Lateral Derived Tables.

Note

LATERAL now is a reserved word and cannot be used as an identifier without identifier quoting.

Thread Pool Notes

• The INFORMATION_SCHEMA tables that accompany the thread pool plugin have been migrated to be available as Performance Schema tables. The INFORMATION_SCHEMA tables now are deprecated and will be removed in a future MySQL version. Applications should transition away from the old tables to the new tables. For example, if an application uses this query:

```sql
SELECT * FROM INFORMATION_SCHEMA.TP_THREAD_STATE;
```

The application should use this query instead:

```sql
SELECT * FROM performance_schema.tp_thread_state;
```

For more information, see Performance Schema Thread Pool Tables.

X Plugin Notes

• X Plugin now includes the 5-digit SQLSTATE error code in its error handling class. Previously, the SQLSTATE error code was returned to clients for SQL errors, but only the MySQL-specific error number was exposed. (Bug #28735058)

• When querying collections of documents, if boolean values were used as arguments for a placeholder in an SQL query, unexpected results were returned. A new translation specialization has now been added for boolean values so that they are handled correctly in this situation. (Bug #28227037)

• X Protocol now always converts retrieved data to the utf8mb4 character set (using the utf8mb4_general_ci collation) before returning it. (Bug #28180155)

• X Protocol now supports SQL prepare functionality.

Functionality Added or Changed

• InnoDB: Disabling the innodb_buffer_pool_in_core_file variable reduces the size of core files by excluding InnoDB buffer pool pages. To use this variable, the core_file variable must be enabled and the operating system must support the MADV_DONTDUMP non-POSIX extension to madvise(), which is supported in Linux 3.4 and later. For more information, see Excluding Buffer Pool Pages from Core Files.

Thanks to Facebook for the contribution. (Bug #27724476, Bug #90144)

• InnoDB: By default, undo logs reside in two undo tablespaces that are created when the MySQL instance is initialized.

Additional undo tablespaces can be created in a chosen location at runtime using CREATE UNDO TABLESPACE syntax.
CREATE UNDO TABLESPACE `tablespace_name` ADD DATAFILE 'file_name.ibu';

Undo tablespaces created using CREATE UNDO TABLESPACE syntax can be dropped at runtime using DROP UNDO TABLESPACE syntax.

DROP UNDO TABLESPACE `tablespace_name`;

ALTER UNDO TABLESPACE `tablespace_name` SET {ACTIVE|INACTIVE};

A STATE column that shows the state of a tablespace was added to the INFORMATION_SCHEMA.INNODB_TABLESPACES table. An undo tablespace must be in an empty state before it can be dropped.

The previously deprecated innodb_undo_tablespaces variable is no longer configurable and will be removed in a future MySQL version.

For more information, see Undo Tables.

- **InnoDB**: InnoDB now supports parallel clustered index reads, which can improve CHECK TABLE performance. This feature does not apply to secondary index scans. The innodb_parallel_read_threads session variable must be set to a value greater than 1 for parallel clustered index reads to occur. The default value is 4. The actual number of threads used to perform a parallel clustered index read is determined by the innodb_parallel_read_threads setting or the number of index subtrees to scan, whichever is smaller.

- **InnoDB**: The ADD DATAFILE clause of the CREATE TABLESPACE statement is now optional, which permits users without the FILE privilege to create tablespaces. A CREATE TABLESPACE statement executed without an ADD DATAFILE clause implicitly creates a tablespace data file with a unique file name.

- **InnoDB**: When the innodb_dedicated_server variable is enabled, the size and number of log files are now configured according to the automatically configured buffer pool size. Previously, log file size was configured according to the amount of memory detected on the server, and the number of log files was not configured automatically.

- **Replication**: When running a group in single-primary mode, in the event of a new primary being elected while there were transactions held in the backlog to be applied, there was a chance that a read operation against the new primary could return a stale value. Now, you can use the group_replication_consistency variable to control how a group behaves in this situation. When group_replication_consistency is set to EVENTUAL, a new primary responds to read requests even when there is a backlog which has not yet been applied, which matches the previous behavior and comes with the risk that a client could read old values while any backlog is being applied. Writes to the new primary fail during this period because it is in super_read_only mode enabled. When group_replication_consistency is set to BEFORE_ON_PRIMARY_FAILOVER, any new read or write queries against a newly elected primary that is applying backlog from the old primary are held until the backlog is applied. This ensures that clients always read the newest value which they have written, but also means that clients might have to wait until the backlog has been applied before they can read from the new primary.

References: See also: Bug #26004894.

- **Microsoft Windows**: The access control granted to clients on the named pipe created by the MySQL server now is set to the minimum necessary for successful communication on Windows. Newer MySQL
client software can open named pipe connections without any additional configuration. If older client software cannot be upgraded immediately, the new `named_pipe_full_access_group server system variable can be used to give a Windows group the necessary permissions to open a named pipe connection. Membership in the full-access group should be restricted and temporary.

- The minimal server RPM is mostly used for Docker images. For better Docker compatibility, the `log-error` line has been removed from the `rpm-docker` configuration file. This way, logging goes to `stdout/stderr`, enabling use of Docker’s own interface. (Bug #28692675)

- Error messages relating to creating and dropping foreign keys were improved to be more specific and informative. (Bug #28526309, Bug #92087)

- The error message for `ALTER TABLE` statements that attempted character set conversion but failed was improved to indicate which column produced the error. (Bug #27546306, Bug #88738)

- Previously, for command options that take a numeric value, the value could be given with a suffix of `K`, `M`, or `G` to indicate a multiplier of `1024`, `1024^2` or `1024^3`. Now a suffix can also be `T`, `P`, and `E` to indicate a multiplier of `1024^4`, `1024^5` or `1024^6`. Thanks to Daniel Black for the patch. (Bug #27306931, Bug #89017)

- Resource group locking was revised to improve scalability and performance. (Bug #27148580)

- The Group Communication System (GCS) and group communication engine (XCom, a Paxos variant) for Group Replication now provide full support for IPv6, so replication group members can use IPv6 addresses as an alternative to IPv4 addresses for internal group communications. The localhost address for IPv6, and the private subnetwork addresses for IPv6 (unique-local addresses and link-local unicast addresses), are added to the automatic whitelist for Group Replication for use if no manual whitelist is specified.

  If all members of a replication group are at a MySQL server version that supports the use of IPv6 addresses for Group Replication, the group can contain a mix of members using IPv6 addresses and members using IPv4 addresses. Joining members must provide whitelisted IP addresses or host names that match the protocols offered by the seed members for connection, but the joining member's main identifying address or host name (`group_replication_local_address`) can use either protocol. If a member uses a host name that resolves to both an IPv4 and an IPv6 address, the IPv4 address is always used for Group Replication connections.

  If any or all existing members of a replication group are using an older MySQL Server version without support for the use of IPv6 addresses for Group Replication, joining members must present an IPv4 address for group communications in the `group_replication_local_address` option. When every group member has been upgraded, the group can be migrated to IPv6 addresses. (Bug #26088469, Bug #27757729, Bug #90217)

- The startup option `--binlog-row-event-max-size` now has a corresponding system variable `binlog_row_event_max_size`. The startup option and system variable set a soft limit on the maximum size of a row-based binary log event, with a default setting of 8192 bytes. Where possible, rows stored in the binary log are grouped into events with a size not exceeding the value of this setting. If an event cannot be split, the maximum size can be exceeded.

  The `binlog_row_event_max_size` global system variable is read-only and can be set only at server startup. Its value can therefore only be modified by using the `PERSIST_ONLY` keyword or the `@@persist_only` qualifier with the `SET` statement. The addition of a system variable means that this setting can be viewed using the Performance Schema tables or a `SHOW VARIABLES` or `SELECT` statement. (Bug #19985377, Bug #74728)

- MySQL Group Replication can now communicate using a dedicated input channel as an alternative to using a TCP socket. The new input channel uses shared memory for communication between the Group
Replication logic and the local instance of the underlying group communication engine (XCom, a Paxos variant).

Previously, communication with the local XCom instance always took place using a TCP socket, namely the network address that is specified by the `group_replication_local_address` system variable for each group member. This incurred overheads that were unnecessary for local communication, such as memory copying through the network protocol stack and data serialization. A TCP socket (`group_replication_local_address`) is still required for each group member to communicate with remote XCom instances. The Group Communication System (GCS) component of Group Replication now selects the most appropriate communication method for each Group Replication task, either the input channel or TCP. For example, the process of joining a group requires communication with a remote XCom instance, so TCP must be used. However, the process of removing a member from a group only requires communication with the local XCom instance, so the input channel is used. The input channel is selected wherever possible to minimize the overheads associated with communication using networking mechanisms.

- Two new session system variables have been added for internal use by replication. `original_server_version` and `immediate_server_version` support cross-version replication by transmitting the MySQL server release numbers associated with a transaction through the replication topology. `original_server_version` holds the MySQL Server release number of the server where a transaction was originally committed (for example, 80014 for a MySQL 8.0.14 server instance). `immediate_server_version` holds the MySQL Server release number of the server that is the immediate master in a replication topology. If either of those servers, or another intervening server in the replication topology, is at an older release that does not support these session system variables, their values are set to 0.

With this information the slave can correctly process data originating from a master at an older release, by recognizing where syntax changes or semantic changes have occurred between the releases involved and handling these appropriately. The information can also be used in a Group Replication environment where one or more members of the replication group is at a newer release than the others. The value of the variable can be viewed in the binary log for each transaction (as part of the `Gtid_log_event`, or `Anonymous_gtid_log_event` if GTIDs are not in use on the server), and could be helpful in debugging cross-version replication issues.

- `ALTER TABLE` now can be used to change a column character set in place (without a table rebuild), when these conditions apply:
  - The column data type is CHAR, VARCHAR, a TEXT type, or ENUM.
  - The character set change is from utf8mb3 to utf8mb4, or any character set to binary.
  - There is no index on the column.
  - The new `-DFORCE_INSOURCE_BUILD` CMake option defines whether to force an in-source build. Out-of-source builds are recommended, as they permit multiple builds from the same source, and cleanup can be performed quickly by removing the build directory. To force an in-source build, invoke CMake with `-DFORCE_INSOURCE_BUILD=ON`.

**Bugs Fixed**

- **Important Change:** Importing a dump from a MySQL 5.7 server to a server running MySQL 8.0 often failed with `ER_WRONG_VALUE_FOR_VAR` when an SQL mode not supported by the 8.0 server was used. This could happen frequently due to the fact that `NO_AUTO_CREATE_USER` is enabled by default in MySQL 5.7 but not supported in MySQL 8.0.
The behavior of the server in such circumstances now depends on the setting of the `pseudo_slave_mode` system variable. If this is false, the server rejects the mode setting with `ER_UNSUPPORTED_SQL_MODE`. If `pseudo_slave_mode` is true, the server ignores the unsupported mode and gives a warning. Note that `mysqlbinlog` sets `pseudo_slave_mode` to true prior to executing any SQL. (Bug #90337, Bug #27828236)

- **InnoDB**: Global and backup metadata locks were not released after the background purge thread truncated undo logs. (Bug #29215254, Bug #93901)

- **InnoDB**: MySQL would not start on Solaris X86. The static thread-local 'tables' variable in the TempTable storage engine was not properly initialized. (Bug #28987365)

- **InnoDB**: Latching logic used during deadlock detection was simplified. (Bug #28904966)

- **InnoDB**: An invalid record offset for an old version of a clustered index record raised a debug assertion. (Bug #28825617)

  References: This issue is a regression of: Bug #25540277.

- **InnoDB**: The minimum DML delay imposed when the length of the history list exceeds `innodb_max_purge_lag` was decreased from 5000 microseconds to 5 microseconds. (Bug #28813453)

- **InnoDB**: An incorrect lock order caused a deadlock when one thread attempted to drop a table while another created an encrypted tablespace. (Bug #28774259)

- **InnoDB**: `ALTER TABLESPACE` failed to ignore unsupported tablespace attributes. (Bug #28656611)

- **InnoDB**: Implicit to explicit lock conversion logic was simplified and optimized. (Bug #28637472)

- **InnoDB**: A fragment page allocation failure raised an assertion. (Bug #28615893)

- **InnoDB**: Incorrectly placed debug points caused flushed LOB pages to be considered corrupt. (Bug #28607368)

- **InnoDB**: The TempTable storage engine incorrectly created temporary files in the system temporary directory instead of the directory defined by the `tmpdir` variable. (Bug #28598943)

- **InnoDB**: Attempting to drop a table with a name similar to that of a full-text search auxiliary table caused an assertion failure. (Bug #28577083)

- **InnoDB**: A function called by an `UPDATE` query did not account for virtual columns. (Bug #28560650)

- **InnoDB**: An incorrect key was defined for the buffer pool zip hash mutex. (Bug #28556539)

- **InnoDB**: Deadlock handling for background transactions that involve the `mysql.innodb_table_stats` and `mysql.innodb_index_stats` tables was modified. The tables were incorrectly included in an assertion that is triggered when internal tables are included in a deadlock cycle. (Bug #28523042, Bug #92069)

- **InnoDB**: Setting `innodb_spin_wait_delay` to a high value caused an assertion failure when attempting to shut down the server. To prevent this failure from occurring, the `innodb_spin_wait_delay` maximum value was reduced to 1000. (Bug #28489407, Bug #91973)

- **InnoDB**: An `ON DELETE CASCADE` operation on table with a foreign key constraint and an indexed virtual column caused the server to exit. (Bug #28470805)

- **InnoDB**: An incorrectly written DML log involving a virtual column value raised an assertion. (Bug #28448853)
• InnoDB: A **RENAME TABLE** operation failed when run on a table created outside of the MySQL data directory using the **DATA DIRECTORY** clause. (Bug #28341514)

• InnoDB: **ALTER TABLE ... EXCHANGE PARTITION** permitted partitions with different virtual column definitions to be exchanged, which resulted in an assertion when InnoDB later attempted to read from a nonexistent virtual column. (Bug #28235668)

• InnoDB: A counter was added for redo log write and flush requests that occur during transaction commit. The counter is used by the log writer thread to compute the average time between consecutive requests. When the average time is greater than 100 microseconds, log writer threads do not use spin delay and instead wait on request events with a 10 microsecond timeout limit.

A log writer thread implementation issue that could cause a hang was also fixed. (Bug #28062382, Bug #28444247, Bug #28616442, Bug #90890)

• InnoDB: An assertion was raised when attempting to add rollback segments to newly added undo tablespace that was not fully initialized. (Bug #27914054)

• InnoDB: Foreign key constraints were ignored after a **RENAME TABLE** operation. (Bug #27453180, Bug #89441)

• InnoDB: Using the **O_DIRECT_NO_FSYNC** innodb_flush_method setting could cause the system to hang due to file system metadata becoming unsynchronized. To prevent this issue from occurring in **O_DIRECT_NO_FSYNC** mode, InnoDB now calls `fsync()` after creating a new file, after increasing file size, and after closing a file. The `fsync()` system call is still skipped after each write operation. (Bug #27309336)

• InnoDB: Specifying the **CREATE TABLE** or **ALTER TABLE ENCRYPTION** option with an empty string failed to raise an error and was interpreted as a default setting, which is **ENCRYPTION='N'**. Specifying an empty string is now treated as invalid and raises an error. (Bug #27177845)

• InnoDB: Partition suffixes (the #P# or #SP# part of a partitioned table name) were not converted to lowercase when moving tablespace data files from a MySQL instance on Windows to a MySQL instance on Linux where the **lower_case_table_names** variable was enabled. Failure to fully convert table names to lowercase caused errors such as "’InnoDB error’ from storage engine" when attempting to alter, rename, or optimize the tables. (Bug #26925260)

• InnoDB: An assertion was raised when attempting to write to a tablespace file greater than 4GB in size on a 64-bit Windows system. The failure was due to a narrowing cast. (Bug #26636815, Bug #87423)

• InnoDB: After attempting to create a table with a foreign key constraint that referenced a partitioned table, which is an unsupported operation, `SHOW ENGINE INNODB STATUS` output incorrectly reported a foreign key error indicating that the referenced table name could not be resolved. This error no longer appears, and the error message returned to the client now states that foreign keys are not yet supported in conjunction with partitioning. (Bug #25319071, Bug #84331)

• Partitioning: Trying to perform an instant add column on a discarded tablespace led to an assert. An error is now returned in such cases. (Bug #28517843)

• Partitioning: Repeated **ALTER TABLE** statements on partitioned tables containing **BLOB** or **TEXT** columns were not always handled correctly. (Bug #28491099)

• Partitioning: **ALTER TABLE ... EXCHANGE PARTITION** did not work when the partitioned table had one or more partition definitions using the **DATA DIRECTORY** option. This fix supports partitioned tables using the InnoDB storage engine only. (Bug #19730200)

• Replication: Depending on the value of **group_replication_exit_state_action**, the behavior of members exiting a group was not consistent. To harmonize the behavior
of members exiting the group regardless of the error scenario, now when a member with 
group_replication_exit_state_action=READ_ONLY exits the group unintentionally, the 
super_read_only mode that the member had when started is restored. This makes the behavior 
consistent with that of a member with group_replication_exit_state_action=ABORT_SERVER. 
(Bug #28971639, Bug #28526591)

• Replication: The metadata written to the binary log for CREATE TABLE statements includes character 
set information for the character columns in the table. Previously, when the mysqlbinlog option --print-table-metadata was specified, a default character set was printed for the table. This default 
character set was the character set that appeared most frequently in the table columns, and might 
not match the default character set that had been specified for the table. mysqlbinlog now prints 
the character set for each column individually. The columns are also printed on separate lines. (Bug 
#28774144)

• Replication: Character set information was not written to the binary log as part of the table metadata 
for ENUM and SET columns. This information is now added when binlog_row_metadata=FULL is set, 
which produces extended metadata. (For character columns, character set information is also added 
with binlog_row_metadata=MINIMAL.) (Bug #28706307)

• Replication: A patch to correct the handling of quotes for identifiers in ROLLBACK TO SAVEPOINT 
statements in the binary log was not correctly applied to subsequent MySQL versions. (Bug #28569645)

• Replication: Following a patch in MySQL 5.7.23, LOAD DATA statements stopped statement-based 
replication from a MySQL 5.7.22 master to a replication slave at a later release. The problem has now 
been fixed. (Bug #28541204, Bug #92132)

• Replication: In some circumstances, the CHANGE MASTER TO statement could not 
be used on a replication slave if the master info log had been changed from a table 
(master_info_repository=TABLE) into a file (master_info_repository=FILE). (Bug 
#28529558)

• Replication: mysqlbinlog incorrectly added statements to set the sql_require_primary_key 
system variable (which was introduced in MySQL 8.0.13) to ON for events involving DML SQL 
statements. The check carried out when the system variable is set to ON is only relevant for DDL SQL 
statements that create new tables or alter the structure of existing tables. (Bug #28524803)

• Replication: When the system variables binlog_transaction_dependency_tracking and 
binlog_transaction_dependency_history_size were set or read, the types of lock that were 
required could result in a deadlock scenario, because the same locks were also required for working with 
the active binary logs. A new lock type is now used instead for access to the transaction dependency 
tracking system variables, so that this deadlock cannot occur. (Bug #28511326, Bug #91941, Bug 
#28537209, Bug #92108)

• Replication: An assertion was raised in debug builds if an implicit commit was 
attempted when the GTID value for the next transaction had not yet been determined 
(gtid_next=NOT_YET_DETERMINED). The gtid_next system variable has this value 
immediately after the internal-use statement BINLOG has been issued by mysqlbinlog 
to execute a format description event. If a statement with an implicit commit was attempted 
next (such as a CREATE TABLE statement), the gtid_next setting did not transition to 
AUTOMATIC state, and was left in an unacceptable state. If autocommit was on, the error 
ER_CANT_SET_GTID_NEXT_TO_ANONYMOUS_WHEN_GTID_MODE_IS_ON was also logged when the 
statement was attempted.

To fix this issue, the use of the BINLOG statement is now prevented during transactions if it would 
change the state of gtid_next. The error ER_VARIABLE_NOT_SETTABLE_IN_TRANSACTION 
is returned if this is attempted. Also, when GTIDs are in use and the value of
**Replication:** The `PURGE BINARY LOGS TO 'log_name'` statement failed for binary log files that had been moved to another location using `mysqlbinlogmove`. Such files are still listed in the binary log index file, but they are listed using an absolute path, rather than a path relative to the directory where the binary log files are normally stored. MySQL Server can now locate and purge moved binary log files successfully. (Bug #28284624)

**Replication:** When `binlog_format` is set to `MIXED`, if a function contained DML statements that applied to a temporary table, and also a `DROP TEMPORARY TABLE` statement, the function call was not written to the binary log, which caused replication errors. The function call is now written to the binary log in mixed replication mode if the function contains DML statements that operate on a temporary table. (Bug #28258992)

**Replication:** If `autocommit` was set to 0 for a replication slave or Group Replication group member where GTIDs were in use and `super_read_only=ON` was set, server shutdown was prevented by a transaction that did not complete. The transaction was attempting to save GTIDs to the `mysql.gtid_executed` table, but the update failed because `super_read_only=ON` was set. (With `autocommit` set to 1, the transaction would complete in this situation, and the `mysql.gtid_executed` table would instead be updated at server startup.) Now, the check for the `super_read_only` setting is skipped for this task, so the transaction is able to save the GTIDs to the `mysql.gtid_executed` table and complete regardless of the combination of `super_read_only` and `autocommit` settings. (Bug #28183718)

**Replication:** An assertion was raised in debug builds if an `XA ROLLBACK` statement was issued for an unknown transaction identifier when the `gtid_next` value had been set manually. The server now does not attempt to update the GTID state if an `XA ROLLBACK` statement fails with an error. (Bug #27928837, Bug #90640)

**Replication:** An assertion was raised in debug builds if a `SELECT... FOR UPDATE` statement was issued immediately after a transaction was committed or rolled back, and the transaction had been assigned a GTID manually using the `gtid_next` session system variable. After `gtid_next` has been used to set a GTID for a transaction, and the transaction has been committed or rolled back, another explicit `SET GTID_NEXT` statement must be issued before any other statement, otherwise the `gtid_next` value is left undefined. The `SELECT... FOR UPDATE` statement caused a GTID consistency violation in this situation because it acquired write locks, although it did not make any changes. `SELECT... FOR UPDATE` statements that acquire write locks now return an error in this situation. (Bug #27903848, Bug #90547)

**Replication:** Under heavy loads, a race condition in binary log group commit could cause the server to stop unexpectedly. The tracking of transaction commits has been changed to prevent this situation. (Bug #27556117)

**Replication:** The value returned by a `SHOW SLAVE STATUS` statement for the total combined size of all existing relay log files (`Relay_Log_Space`) could become much larger than the actual disk space used by the relay log files. The I/O thread did not lock the variable while it updated the value, so the SQL thread could automatically delete a relay log file and write a reduced value before the I/O thread finished updating the value. The I/O thread then wrote its original size calculation, ignoring the SQL thread's update and so adding back the space for the deleted file. The `Relay_Log_Space` value is now locked during updates to prevent concurrent updates and ensure an accurate calculation. (Bug #26997096, Bug #87832)
• **Replication:** If the relay log index file was temporarily locked for viewing by a backup process for a replication slave, and MySQL Server also attempted to access the file at that time for rename or delete operations, the backup completed with warnings, but MySQL Server experienced an unexpected halt. MySQL Server now retries the file access operation a number of times in case this or a similar scenario is the explanation and the file becomes available again before long. (Bug #25839610)

• **Replication:** With `sync_binlog=1` set, if the binary log was rotated during a commit before the binary log end position was updated, replication stopped on the slave because the server attempted to use the old binary log end position with the new binary log file. The server now compares the binary log file name with the active binary log file when updating the binary log end position, so that the issue does not occur. (Bug #22252394, Bug #25524203, Bug #84752)

• **Replication:** When adding a new member to a group, if the certification information was too big to transmit, an event was generated that caused failures in all group members. To avoid this situation, now if the certification information is too large an error is generated which makes the joining member leave the group. (Bug #93130, Bug #91870, Bug #28900691, Bug #28443958)

• **Replication:** When a group was being reconfigured online, for example using `group_replication_switch_to_multi_primary_mode` or `group_replication_set_as_primary`, there was a chance that stopping a member could result in an unexpected stop. Now, when you issue `STOP GROUP_REPLICATION`, if the member is part of an online group that is being reconfigured, the group coordinator is informed that Group Replication is stopping. The member waits for the online configuration process to complete any ongoing actions, but any subsequent actions are cancelled. (Bug #92829, Bug #28807260)

• **Replication:** When stopping replication, any channels that had pending transactions could cause a deadlock in Group Replication. (Bug #92376, Bug #28636768, Bug #28365855)

• **Replication:** When `group_replication_exit_state_action` is set to `ABORT_SERVER`, the Group Replication plugin now uses the new component service added by WL#12003 to shutdown MySQL. (Bug #91793, Bug #28401703)

• **Replication:** When you used `group_replication_switch_to_single_primary_mode()`, if a member which also had an asynchronous channel encountered an error, the asynchronous replication channel was not stopped correctly, and the server could stop unexpectedly. (Bug #91747, Bug #28382590)

• **Replication:** It was possible to use the group coordinator based UDFs which configure a group, such as `group_replication_switch_to_single_primary_mode`, while members were in the `UNREACHABLE` or `RECOVERING` state, and this caused the operation to wait until all members became `ONLINE`. This could result in the group coordinator operation never completing successfully. Now, if you issue any of these UDFs on a group in this state, an error is returned. Ensure all members are `ONLINE` before attempting to configure the group using the UDFs. (Bug #91537, Bug #28284355)

• **Replication:** When a member joined a group that had a constant peak load, the member might not be able to move from the `RECOVERING` to the `ONLINE` state. The cause was that:

  • the member was waiting in a loop for the complete queue of transactions that arrived during recovery to be applied, while new transactions were still arriving.

  • even when the complete queue had been applied, the member was also checking that the applier was paused, which is unlikely to happen in a continuous peak workload.

Now, when the recovery completion policy is waiting for transactions to be applied, the member first waits until one of the following conditions is fulfilled:
MySQL 8.0 Release Notes

- the transactions to apply fit within the flow control configuration. In other words, the transactions to be applied can be applied during the next flow control iteration;

- no transactions are being queued or applied, in the case of an empty recovery queue.

Then, the member waits for the currently queued transactions in the `group_replication_applier` channel to be applied, before the member state changes to ONLINE. (Bug #89582, Bug #27511404)

**Microsoft Windows:** MySQL Installer could fail after failure to remove an existing MySQL service. This is now treated as nonfatal so that installation operations can continue, but might require a system restart to permit service cleanup. (Bug #29016677, Bug #93048)

**Microsoft Windows:** When multiple instances of `mysqld` were started with the `--no-monitor` option on the same host for same user, the `SHUTDOWN` command shut down the wrong server process. This fix creates a unique shutdown event name for use with `--no-monitor` by appending the process ID of the process. (Bug #28723675)

**X DevAPI:** When using the X Protocol, a stored procedure called with a user variable as an OUT parameter did not set the variable’s value. (Bug #91907, Bug #28458752)

**JSON:** Iteration over JSON objects resulted in unnecessary allocation of strings. (Bug #28975640)

**JSON:** Conversion of JSON values to text caused linear growth of the destination string, resulting in an unnecessarily high number of reallocations. Now this process uses exponential growth instead, to reduce the number of allocations required. (Bug #28949700)

**JSON:** `YEAR` values were stored as opaque data in JSON; when JSON documents containing `YEAR` values were converted to text, the `YEAR` values were shown as base64-encoded strings. To resolve this issue, `YEAR` values are now stored as unsigned integers, which are shown as numbers when converted to text. An additional benefit of this fix is that less storage space is now required for `YEAR` values within JSON documents. (Bug #28947107)

**JSON:** Hit an assert when attempting to execute UPDATE or DELETE on an ARCHIVE table containing a JSON column. (Bug #28923281)

**JSON:** When trying to select from a JSON column of a FEDERATED table, the server returned `ER_INVALID_JSON_PATH_CHARSET Cannot create a JSON value from a string with CHARACTER SET 'binary'`. In addition, neither DELETE or UPDATE had any effect on a FEDERATED table containing a JSON column. (Bug #28877215)

**JSON:** A query of the form `SELECT jt.* FROM t1, JSON_TABLE(t1.c, '$[*]' COLUMNS (num INT PATH '$[0]')) AS jt` failed due a permissions error even though the user executing the query had the SELECT privilege on column c. (Bug #23254268)

The code contributed by Facebook for the feature implemented by Bug#27855592 was updated. (Bug #28950397)

References: See also: Bug #27855592.

- On SuSE Linux, spurious EBUSY return values from `pthread_mutex_destroy()` were not handled. (Bug #28948462)
- `mysqld_safe` and `mysqld_multi` were incorrectly included in client-only packages. (Bug #28942508)
- Mishandling of host cache locking could cause a server exit. (Bug #28936159)
MySQL 8.0 Release Notes

- MySQL Enterprise Firewall did not work well if the `audit_log` plugin was installed. (Bug #28930885, Bug #93184)
- Corrections were made to enable successful builds under Visual Studio on Windows. (Bug #28892711, Bug #93077)
- The server permitted creation of databases with the same name as redo log files, which could result in unexpected server behavior. Such names are no longer permitted as database names. (Bug #28867993)
- `mysqld_multi` could fail to pass the correct `datadir` value to `mysqld`. (Bug #28866662, Bug #90801)
- A debug assertion that checks parameter schema names during MDL key creation for routines, events, and triggers to ensure that names are lowercase failed when encountering a schema name that included multi-byte character. (Bug #28864244)
- The format specifiers for some error messages were improved to avoid displaying incorrect numeric values. (Bug #28860795)
- For debug builds on Windows, unused memory leak checks were enabled and could slow down the shutdown process. These checks are now enabled only for specialized builds. (Bug #28857626)
- `mysql_upgrade` could fail to upgrade certain system tables if the `sql_require_primary_key` system variable was enabled. (Bug #28855207, Bug #92988)
- Builds configured with `-DWITH_LIBWRAP=ON` did not compile. (Bug #28853650, Bug #92983)
- For InnoDB tables, the values of stored or indexed virtual generated columns that depended on the `DEFAULT()` function were not correctly updated by `ALTER TABLE`, if the default for a column referenced in this function was changed by making column nullable. (Bug #28848265)
- Corrections were made to enable successful builds under Visual Studio on Windows with the `/permissive` flag turned on. (Bug #28842878, Bug #92943)
- Builds configured with `-DCMAKE_BUILD_TYPE=Release` did not compile. (Bug #28841366, Bug #92945)
- `ALTER TABLE` now can use the `INPLACE` algorithm when these conditions apply:
  - For InnoDB tables, statements that modify generated stored columns but do not change their type, expression, or nullability.
  - For non-InnoDB tables, statements that modify generated stored or virtual columns but do not change their type, expression, or nullability.
  
  An example of such a change is a change to the column comment. (Bug #28836543)
- Plugin system variables that had been persisted were not applied when the plugin was reinstalled. (Bug #28823972)
- `EXPLAIN ... FOR CONNECTION` could modify another connection's SQL mode. (Bug #28786981)
- Removal of Sun RPC and XDR from `glibc` into a separate `libtirpc` library caused problems with `libasan` on some platforms. (Bug #28785835, Bug #92762, Bug #28897799, Bug #93116)
- It was possible to hit an assert when comparing two `ENUM` values while processing a query of the form `SELECT a FROM table WHERE b = value` and there was an index on column `b`. (Bug #28769996)
- Concurrent read and write access to the `offline_mode` system variable could result in deadlock. (Bug #28761869)
• Triggers were loaded into memory in an incorrect order when upgrading from MySQL 5.7 to MySQL 8.0, causing an assertion failure. (Bug #28760011, Bug #92609)

• Joins involving the Performance Schema `data_locks` table could produce incorrect results. (Bug #28733170)

• Some multiply-nested subqueries involving the use of scalar subqueries were not handled correctly. (Bug #28723670)

• On Ubuntu, the installed `/etc/mysql/mysql.conf.d/default-auth-override.cnf` file was mistakenly created with executable mode. Thanks to Evgeniy Patlan for the correction contribution. (Bug #28714840, Bug #92587)

• A memory leak was caused by `GET_LOCK()` calls with a zero timeout that failed due to concurrent connections holding the same user-level lock. (Bug #28714367)

• Heap corruption and a server exit could occur when a server hosting a large number of tables was started and stopped repeatedly. (Bug #28705511, Bug #92572)

• MySQL Router was missing from MySQL Server MSI packages. (Bug #28685556)

• The example stored function `GTID_SUBTRACT_UUID` has been corrected in the code to match the documented version. (Bug #28670170)

• `CAP_SYS_NICE` capability is no longer enabled for `mysqld` by MySQL package installers for Linux. (This was done to facilitate use of resource groups thread priorities.) For Linux deployments that require access to thread priorities, consult the MySQL Reference Manual instructions for enabling `CAP_SYS_NICE` capability at Resource Group Restrictions. (Bug #28670160)

• The internal implementation of the `<>` operator was simplified. (Bug #28660232)

• After a `STOP GROUP_REPLICATION` statement was issued to remove a server instance from a group, multiple instances of the error message "[GCS] Error pushing message into group communication engine" were logged on the server instance. The error is now ignored when a server is in the process of leaving a group or is no longer a member of a group. (Bug #28658228, Bug #92454)

• The maximum timeout setting for the waiting period before expelling a suspected Group Replication group member has been reduced to 3600 seconds (one hour). Previously, the `group_replication_member_expel_timeout` system variable could be set to a value of up to 31536000 seconds. The new upper limit provides a more reasonable maximum for the removal of inactive members from the group. The default setting for the timeout is zero, meaning that inactive members are liable for expulsion immediately after the 5-second detection period ends. Specifying a timeout value is useful to avoid unnecessary expulsions on slower networks, or in the case of expected transient network failures or machine slowdowns. (Bug #28656750)

• If the `CHARACTER SET` attribute of some column was implicit in `JSON_TABLE(... COLUMNS ...)`, the resulting column used the global `character_set_results` as the default character set. The column now uses the session `character_set_connection` and `collation_connection` values. (Bug #28643862)

• Adding a functional index on an expression that produced a row value raised an assertion; now it results in an error instead. (Bug #28643252)

• In debug builds, creating a trigger after setting `sql-mode` to `TIME_TRUNCATE_FRACTIONAL` caused an assertion failure. The SQL mode was not present in the `sql_mode` column of the `mysql.triggers` data dictionary table. (Bug #28642918)

• When using `--log-timestamps=SYSTEM`, ISO 8601 timestamps in log messages did not take account of daylight saving time. (Bug #28632725)
• The arguments for error ER_IB_MSG_720 were computed incorrectly. (Bug #28629175)

• The server could exit at startup if the option for specifying a socket file was not specified correctly. (Bug #28609181)

• It was possible to create an inconsistent foreign key by adding a parent table with a different storage engine from the child table, then changing the parent table to the same storage engine as the child table. (Bug #28608460, Bug #92317)

• When a server is joining a replication group, it attempts to connect to the first seed member listed in its group_replication_group_seeds system variable. If the connection is refused, the joining member tries to connect to each of the other seed members in the list in order. Previously, if the joining member connected to a seed member but did not get added to the replication group as a result, the joining member did not make any further connection attempts. This situation could occur if the seed member failed after the connection was made, or if the seed member did not have the joining member’s address in its whitelist and closed the connection, or if the seed member rejected the joining member’s request to join the group. Now, if the joining member connects to a seed member but does not manage to join the group, the joining member continues to try the remaining seed members in the list in order. (Bug #28602835)

• Given certain patterns of allocations, copies with rebinds of the allocator, and deallocations, it was possible for temptable::Allocator to reuse a freed memory block. This led to failures in the test suite on Windows platforms. (Bug #28595557)

• Setting time_zone to a negative offset and timestamp to a low value triggered an assertion when altering routines and views. (Bug #28590623, Bug #92273)

• Persisting the pid_file system variable to DEFAULT could result in a value of NULL for subsequent server startups. (Bug #28589736)

• Incorrect privilege checking could produce an error for SELECT ... FOR UPDATE statements that executed successfully in MySQL 5.7. (Bug #28581664, Bug #92254)

• Attempting to rename the parent column of a foreign key with ALTER TABLE could fail. (Bug #28581468)

• Privileges for RESET PERSIST were not checked correctly. (Bug #28564239)

• An overflow occurred when calculating AVG(YEAR(datetime_column)). (Bug #28562930)

• After a server restart, path names of persisted system variables in the Performance Schema variables_info table could be calculated incorrectly. (Bug #28561584)

• A partitioned table name check raised an invalid assertion. (Bug #28556942)

• The handler::create() function could be called with an error in the condition list, which could prevent an error in the handler::create() function from being reported properly. (Bug #28556264)

• For ALTER TABLE, ALGORITHM=INSTANT was incorrectly rejected on tables created in a MySQL version prior to 8.0.12. (Bug #28554157, Bug #92194)

• mysqlpump did not free all allocated resources when it encountered an error, resulting in memory leaks. (Bug #28538971, Bug #92131)

• The COLLATE attribute was rejected for data types in the COLUMNS clause of the JSON_TABLE() function. (Bug #28538315)

• For debug builds, the server could exit when attempting to roll back CREATE USER statements. (Bug #28536312)

• Plugin variables with signed values were displayed incorrectly. (Bug #28534414, Bug #92107)
• Mishandling of deprecated system variables could cause output from queries on the Performance Schema variables_by_thread table to be incorrect. (Bug #28515475, Bug #92049)

• Data races discovered by Thread Sanitizer in Event_queue::lock_data and the SAFE_MUTEX implementation were fixed. (Bug #28510721, Bug #92041, Bug #28510691, Bug #92040)

• No ER_NEED_REPREPARE diagnostic was pushed to the diagnostics area when a reprepare failed for prepared statements. (Bug #28509306, Bug #92029)

• When a subquery contained a UNION, the count of the number of subquery columns was calculated incorrectly. (Bug #28499924)

• When evaluating an expression using WITH ROLLUP, we now write the result of the expression into a temporary table only when it has a temporary table column. (Bug #28493849, Bug #28523014)

• For debug builds, incorrect foreign key error checking for ALTER TABLE on a TEMPORARY table could result in a server exit. (Bug #28493257, Bug #91990)

• For some system variables, SET PERSIST persisted the default value rather than the specified value. (Bug #28466045)

• SET RESOURCE GROUP could not be executed as a prepared statement. (Bug #28448258, Bug #91876)

• Restored a call to Item_field::fix_fields() that was inadvertently removed during work done to implement window functions. (Bug #28431783)

• Data races reported by Thread Sanitizer during X Plugin startup and shutdown were corrected. (Bug #28407294)

• Creating a table with a partition description containing illegal utf8 characters raised an assertion. (Bug #28387488, Bug #91763)

• mysqldump output could include SQL mode values that have been removed. (Bug #28373001, Bug #91714)

• A potential lock order cycle was corrected. (Bug #28366531)

• On a GTID-enabled server, concurrent statements on the INFORMATION_SCHEMA.COLUMNS table could deadlock. (Bug #28293047, Bug #91548)

• CREATE TABLE statements for tables with the utf32 table character set and literal strings in the table definition raised an assertion. (Bug #28275881)

• Internal functions were added to support updating the server version number upon the successful completion of a server upgrade. (Bug #28211486, Bug #91323)

• Comparing log file names as strings using the memcmp() function resulted in uninitialized memory read errors. The comparison now uses the strncmp() function. Thanks to Zsolt Parragi and Laurynas Biveinis for their contributions. (Bug #28178776, Bug #90238)

• On systems where Group Replication’s Group Communication System (GCS) used the systemd-resolved service for network name resolution, if the host name could not be resolved, GCS kept trying indefinitely. Now, if a retry message is returned from any name resolution service, GCS makes a limited number of retries, then concludes that the host name is unresolvable. (Bug #28177861)

• The server mishandled stored program and resource group names that differed only in accents. (Bug #28122841)

• The optimizer skipped the second column in a composite index when executing an inner join with a LIKE clause against the second column. (Bug #28086754)
MySQL 8.0 Release Notes

- **CREATE TABLE ... SELECT** could create date columns with “zero” date default values when it should have created them without a default value. (Bug #28022129)

- The transformation of **IN** subquery predicates into semijoins was not handled correctly for a very large number of tables. (Bug #28004674)

- The bitmap used for reading fields from the storage engine was not enabled correctly when performing the file sort which is added to the last of any temporary tables created for window functions. In the case where there was no need for a temporary table, the server added a file sort to the output from the select table, but the removed reference (the **WHERE** condition) was not added. Now in such cases, the reference is added to the select table when the first window function needs sorting and no temporary table was created before processing this window function. (Bug #27975193)

- After seeing a row in the range frame, if another row was later determined to be appearing before this range frame, the server continued to check for new rows. This led to the next frame calculation being done incorrectly. (Bug #27973860)

- Server mishandling of **SIGHUP** signals could result in a server exit. (Bug #27966483, Bug #90742)

- **DELETE WHERE a=constant** from a table with column **a** and partitioned by the value of a generated column **b** led to an assertion in debug builds. (Bug #27954073)

- **INFORMATION_SCHEMA** queries could cause a server exit when updating dynamic table statistics. (Bug #27898108)

- A metadata locking deadlock could occur when opening a foreign key parent table. (Bug #27859086)

- Improper memory handling by account management statements could result in server misbehavior. (Bug #27820277)

- In certain cases, window functions did not handle **ORDER BY** and **PARTITION BY** correctly. (Bug #27816506)

- The MySQL query optimizer identifies each predicate to be pushed down to a table as a table condition; as part of this process it checks to see whether a given predicate among the table conditions is already known to be true in virtue of the selected access path for the table, in which case the predicate can safely be eliminated.

  For example, when executing **SELECT * FROM t1 WHERE pk=1**, where **pk** is the primary key of table **t1**, the **ref** access method is selected. Since we know that this already returns only rows for which **pk=1**, further evaluation of this condition as a filter (**Using where**) should be eliminated.

  When optimizing a query which includes a **GROUP BY** or **ORDER BY**, a late optimizer check is performed to discover whether sorting can be skipped by using a sorted index instead. Since this is done after another index may have been chosen for accessing the table to be sorted, some predicates thought to be redundant (due to the previous access path selected) could be removed prematurely. To compensate for this, the following actions were performed:

  - Reconstruction of a table condition containing predicates previously eliminated due to the access method already being selected.
  - Performing a check to see whether any sorted index existed such that sorting could be avoided, possibly modifying the access plan.

  Issues arose because the following actions also intended to remedy early predicate removal were not performed correctly:
• Whether the access plan was modified or not, any extra predicates added back in the reconstructed table condition mentioned previously became a permanent part of the table condition.

• When the access plan was changed to use another sorted index, no analysis was performed for the new index in order to remove predicates made obsolete by the new index.

A further problem existed for storage engines implementing condition pushdown, such as `NDBCLUSTER`:
The conditions pushed down were generated from the table condition prior to analysis, such that, if the access path was later changed, the pushed condition did not contain the predicates already removed, making condition pushdown less efficient.

The root cause of this issue was that `part_of_refkey()` analysis was performed on table predicates before the access method for the table had been completely decided. This is fixed by removing such early analysis. (Bug #27808758, Bug #27814026)

• A windowing function that included an `ORDER BY column` clause failed with `Unknown field in window order by` even when the column was found in the table being queried. (Bug #27808099)

• Executing a prepared statement to do a multiple-row insert with large number of placeholders consumed excessive memory and could execute slowly. (Bug #27703912)

• On Windows, if the Visual C++ Redistributable for Visual Studio had been removed, MySQL uninstallation using the MSI installer failed. (Bug #27621546)

• The parser accepted invalid `SET` statement syntax in trigger definitions that could result in a server exit. (Bug #27595603)

• The server failed to start if the `keyring_encrypted_file` plugin keyring file was invalid. (Bug #27588064)

• Keyring migration failed with source and destination keyring plugins of `keyring_okv` and `keyring_encrypted_file`, respectively. (Bug #27493970)

• It was possible in debug builds for a windowing function using a signed integer to mishandle a frame that included `FOLLOWING`. (Bug #27452365)

• When executing a prepared statement with a procedure call with the `CURSOR_TYPE_READ_ONLY` flag set, the client library hung if the procedure performed a `SELECT` that returned an empty result set. (Bug #27443252, Bug #89214)

• Names of referenced columns of foreign keys were always shown in lowercase in `SHOW CREATE TABLE` output and the `INFORMATION_SCHEMA.KEY_COLUMN_USAGE` table. (Bug #27353767, Bug #88718)

• Loading and unloading the `audit_log` plugin while performing other concurrent activity could cause the server to become unresponsive. (Bug #27325622)

• The data dictionary properties interface (`.dd::Properties`) and implementation was revised to provide a new method of defining valid keys for property objects. (Bug #27309072, Bug #89031, Bug #27309082, Bug #89032)

• Installing and uninstalled the `validate_password` component concurrently with `SET PASSWORD` could cause component failure. (Bug #27020979)

• Some typos in server source code were fixed. Thanks to Hyunwoo Park for the contribution. (Bug #26189673, Bug #86565)
• After column privileges were granted to a table, a `HANDLER READ` call asserted during privilege checking. (Bug #25987758)

• A check that ensures compatibility of referencing and referenced column types in a foreign key definition was moved from the storage engine layer to the SQL layer. In addition, a better error message is produced when columns are not compatible. (Bug #25722927, Bug #28371394, Bug #91712, Bug #21308781, Bug #77467)

• The parser performed some out-of-memory checks incorrectly. (Bug #25633994)

• A race condition between user-management statements and other statements that tried to access grant tables directly could result in deadlock and transaction rollback. (Bug #24481240)

• When the server was started with the `--skip-name-resolve` option, spurious warnings could be written to the error log about ignoring accounts with a host name part of `localhost`. (The accounts in fact were used and not ignored.) (Bug #23329861, Bug #81441)

• DML statements using `IGNORE` were not always handled correctly on tables having generated columns. (Bug #22990029)

• MySQL now removes trivial `WHERE` conditions arising from constant literal expressions during preparation, rather than at a later stage in optimization. This should result in improved plans for queries with outer joins containing trivial conditions, such as this one:

```sql
SELECT * FROM t1 LEFT JOIN t2 ON condition_1 WHERE condition_2 OR 0 = 1
```

After removing the redundant `OR 0 = 1` condition the optimizer can rewrite the query as an inner join, as shown here:

```sql
SELECT * FROM t1 LEFT JOIN t2 WHERE condition_1 AND condition_2
```

For more information, see What Is New in MySQL 8.0, and Outer Join Optimization. (Bug #16893426, Bug #28237111, Bug #28239008, Bug #28341790)

References: See also: Bug #28197977, Bug #28240054.

• Updates for `BLOB` columns in `FEDERATED` tables did not work. (Bug #11748067, Bug #34997)

• Each of the functions `REGEXP_REPLACE()`, `REGEXP_SUBSTR()`, `REGEXP_LIKE()`, and `REGEXP_INSTR()` returned a `DOUBLE` instead of a value of the function's specified return type. (Bug #90039, Bug #27682225)

• A query employing a dynamic range and an index merge could use more memory than expected. (Bug #89953, Bug #27659490)

• Selecting from a table having a `CHAR` column with a `NO_PAD` collation yielded inconsistent results. (Bug #89753, Bug #27578340)

**Changes in MySQL 8.0.13 (2018-10-22, General Availability)**

- Account Management Notes
- Compilation Notes
- Configuration Notes
- Data Type Notes
Account Management Notes

- It is now possible to require that attempts to change an account password be verified by specifying the current password to be replaced. This enables DBAs to prevent users from changing a password without proving that they know the current password. It is possible to establish password-verification policy globally using the `password_require_current` system variable, as well as on a per-account basis using the `PASSWORD REQUIRE` option of the `CREATE USER` and `ALTER USER` statements. Together with existing password-management capabilities, the new capability of requiring verification provides DBAs more complete control over password management. For more information, see Password Management.

  Important

  The implementation of password-verification capability involves a change to the structure of the `mysql.user` system table. If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this system database change. Until this is done, password changes are not possible.

Compilation Notes

- Solaris: MySQL now can be compiled on Solaris using `gcc`. (Bug #27802681)

Configuration Notes

- The new `WITH_LTO` CMake option controls whether to enable link-time optimization. Currently, this is supported only by GCC 7 and 8. (Bug #28184537, Bug #28211382)
• The new `WITH_RAPIDJSON CMake` option controls whether to compile with the bundled or system RapidJSON library. (Bug #28024992, Bug #90867)

• The `CMAKE_BUILD_TYPE CMake` option now supports a `Release` build type, which is like the `RelWithDebInfo` build type but omits debugging information to reduce the build size. (Bug #27874068)

• The new `sql_require_primary_key` system variable makes it possible to have statements that create new tables or alter the structure of existing tables enforce the requirement that tables have a primary key. Enabling this variable helps avoid performance problems in row-based replication that can occur when tables have no primary key. Suppose that a table has no primary key and an update or delete modifies multiple rows. On the master server, this operation can be performed using a single table scan but, when replicated using row-based replication, results in a table scan for each row to be modified on the slave. With a primary key, these table scans do not occur. (Bug #17468242, Bug #69845, Bug #17005592, Bug #69223)

• To enable the server to listen on a set of addresses, the `bind_address` system variable now permits a list of comma-separated IP addresses or host names, not just a single address or name. For details, see Server System Variables.

Data Type Notes

• MySQL now supports use of expressions as default values in data type specifications. This includes the use of expressions as default values for the `BLOB`, `TEXT`, `GEOMETRY`, and `JSON` data types, which previously could not be assigned default values at all. For details, see Data Type Default Values.

Deprecation and Removal Notes

• InnoDB; Partitioning: Support for placing table partitions in shared tablespaces was removed. Shared tablespaces include the system tablespace and general tablespaces. For information about identifying partitions in shared tablespaces and moving them to file-per-table tablespaces, see Preparing Your Installation for Upgrade.

• InnoDB: Support for `TABLESPACE = innodb_file_per_table` and `TABLESPACE = innodb_temporary` clauses with `CREATE TEMPORARY TABLE` is deprecated and will be removed in a future MySQL version.

• The `utf8mb3` character set is deprecated and will be removed in a future MySQL version. Please use `utf8mb4` instead.

• Nested comments have never been supported (although they might be permitted under some conditions), but now are considered deprecated and will be removed in a future MySQL version.

• The deprecated `metadata_locks_cache_size` and `metadata_locks_hash_instances` system variables were removed.

• The `PAD_CHAR_TO_FULL_LENGTH` SQL mode has been deprecated and will be removed in a future MySQL version.

Error Handling

• The MySQL client library now returns better error messages for OpenSSL errors. Thanks to Facebook for the patch. (Bug #27855668, Bug #90418)

• Previously, the `ER_NO_REFERENCED_ROW_2` and `ER_ROW_IS_REFERENCED_2` error messages for foreign key operations were displayed and revealed information about parent tables, even when the user had no parent table access privileges. Error handling for this situation has been revised:
• If the user does have table-level privileges for all parent tables, `ER_NO_REFERENCED_ROW_2` and `ER_ROW_IS_REFERENCED_2` are displayed, the same as before.

• If the user does not have table-level privileges for all parent tables, more generic error messages are displayed instead (`ER_NO_REFERENCED_ROW` and `ER_ROW_IS_REFERENCED`).

An exception is that, for stored programs defined to execute with `DEFINER` privileges, the user against which privileges are assessed is the user in the program `DEFINER` clause, not the invoking user. If that user has table-level parent table privileges, parent table information is still displayed. In this case, it is the responsibility of the stored program creator to hide the information by including appropriate condition handlers. (Bug #19477611)

**INFORMATION_SCHEMA Notes**

• These new INFORMATION_SCHEMA tables are available as views on data dictionary tables:
  - `VIEW_ROUTINE_USAGE` provides information about stored functions used in view definitions.
  - `VIEW_TABLE_USAGE` provides information about tables and views used in view definitions.

For more information, see The INFORMATION_SCHEMA VIEW_ROUTINE_USAGE Table, and The INFORMATION_SCHEMA VIEW_TABLE_USAGE Table.

**Logging Notes**

• **Incompatible Change:** The system variables have been removed that previously configured error logging to the system log (the Event Log on Windows, and `syslog` on Unix and Unix-like systems). Where appropriate, the removed system variables were replaced with new system variables managed by the `log_sink_syseventlog` error log component. The following table shows the old and new variable names.

<table>
<thead>
<tr>
<th>Old System Variable</th>
<th>New System Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>log_syslog_facility</code></td>
<td><code>syseventlog.facility</code></td>
</tr>
<tr>
<td><code>log_syslog_include_pid</code></td>
<td><code>syseventlog.include_pid</code></td>
</tr>
<tr>
<td><code>log_syslog_tag</code></td>
<td><code>syseventlog.tag</code></td>
</tr>
<tr>
<td><code>log_syslog</code></td>
<td>None</td>
</tr>
</tbody>
</table>

Important

Installations that used the old system variable names must update their configuration to use the new variable names. For more information, see Error Logging to the System Log.

References: See also: Bug #27534089.

• A new system variable, `log_error_suppression_list`, specifies which diagnostics should not be written to the error log when they occur with a severity of WARNING or INFORMATION. For example, if a particular type of warning occurs frequently but is not of interest (and thus may be considered undesirable “noise” in the error log), it now can be suppressed.

• The code to handle rewriting of account-management statements was refactored to make it easier to maintain and extend. This work resulted in a few minor user-visible effects in the audit, general, and slow query logs:
• Plaintext passwords are replaced by `<string>` rather than `'<string>'`.
• Default clauses are not written unless specified by the user.
• Previously, messages written to the error log by several internal server methods were logged using the ER_LOG_PRINTF_MSG error code. Each of these messages now is logged using a unique error code.

Optimizer Notes

• The optimizer now supports a Skip Scan access method that enables range access to be used in previously inapplicable situations to improve query performance. For more information, see Skip Scan Range Access Method. Thanks to Facebook for the patch on which this access method is based. (Bug #26976512, Bug #88103)

• MySQL now supports creation of functional index key parts that index expression values rather than column values. Functional key parts enable indexing of values that cannot be indexed otherwise, such as JSON values. For details, see CREATE INDEX Syntax.

• Performance of `SELECT COUNT(*) FROM tbl_name` queries for InnoDB tables was improved for single-threaded workloads and when no extra clauses such as `WHERE` or `GROUP BY` are used.

Packaging Notes

• Binary packages that include curl rather than linking to the system curl library now use curl 7.60.0 rather than 7.45.0. (Bug #28043702)

• Debian packaging was updated to reflect the removal of yaSSL and that OpenSSL is the default SSL library for all builds. (Bug #28025599)

• Test plugins have been moved from server packages to test packages. (Bug #27860172)

• MySQL Router is now included in the MySQL server source and monolithic binary packages.

Performance Schema Notes

• A new Performance Schema stage, `waiting for handler commit`, is available to detect threads going through transaction commit. Thanks to Facebook for the patch. (Bug #27855592, Bug #90417)

References: See also: Bug #28950397.

Pluggable Authentication

• Microsoft Windows: On Windows, MySQL Enterprise Edition distributions now bundle the Cyrus SASL library files `libsasl.dll` and `saslSCRAM.dll` so that the LDAP authentication plugins can use the SCRAM-SHA-1 authentication method.

Plugin Notes

• Previously, MySQL plugins could be written in C or C++. MySQL header files used by plugins now contain C++ code, which means that plugins must be written in C++, not C. (Bug #87725, Bug #26781567)

Security Notes

• MySQL Enterprise Edition now provides data masking and de-identification capabilities, implemented as a plugin library containing a plugin and a set of user-defined functions. Data masking hides sensitive
information by replacing real values with substitutes. MySQL Enterprise Data Masking and De-Identification functions enable masking existing data using several methods such as obfuscation (removing identifying characteristics), generation of formatted random data, and data replacement or substitution. For example:

```sql
mysql> SET @ssn = gen_rnd_ssn();
mysql> SET @masked_ssn1 = mask_ssn(@ssn);
mysql> SET @masked_ssn2 = mask_outer(mask_inner (@ssn, 4, 5, 'A'), 3, 0, 'B');
mysql> SELECT @ssn, @masked_ssn1, @masked_ssn2;
```

+-------------+--------------+--------------+
<table>
<thead>
<tr>
<th>@ssn</th>
<th>@masked_ssn1</th>
<th>@masked_ssn2</th>
</tr>
</thead>
<tbody>
<tr>
<td>980-31-2838</td>
<td>XXX-XX-2838</td>
<td>BBB-AA-2838</td>
</tr>
</tbody>
</table>
+-------------+--------------+--------------+

For more information, see MySQL Enterprise Data Masking and De-Identification.

**Spatial Data Support**

- **Incompatible Change:** Previously, `ST_Area()` supported only geometry arguments having a Cartesian spatial reference system (SRS) and produced an error when invoked with geometry arguments that specified a geographic SRS. `ST_Area()` now supports geometry arguments having a geographic SRS and returns the geodetic area in square meters.

  ![Note](image)

  If spatial data contains geometry values that now are interpreted differently by `ST_Area()`, existing queries using this function will return different results, compared to previous MySQL versions.

- The parser for spatial reference system (SRS) definitions in `CREATE SPATIAL REFERENCE SYSTEM` statements now is stricter about rejecting invalid constructs.

  The OGC 01-009 WKT grammar does not make `AXIS` clauses mandatory in `GEOGCS` spatial reference system definitions. However, if there are no `AXIS` clauses, MySQL cannot determine whether a definition has axes in latitude-longitude order or longitude-latitude order. MySQL now enforces the nonstandard requirement that each `GEOGCS` definition must include two `AXIS` clauses. One must be `NORTH` or `SOUTH`, and the other `EAST` or `WEST`. The `AXIS` clause order determines whether the definition has axes in latitude-longitude order or longitude-latitude order. (Bug #28186073, Bug #28147723)

- Previously, `ST_Validate()` supported only geometry arguments having a Cartesian spatial reference system (SRS) and produced an error when invoked with geometry arguments that specified a geographic SRS. `ST_Validate()` now supports geometry arguments having a geographic SRS.

  ![Note](image)

  If spatial data contains geometry values that now are interpreted differently by `ST_Validate()`, existing queries using this function will return different results, compared to previous MySQL versions.

- MySQL now implements the `ST_Transform()` spatial function for use in converting geometry values from one spatial reference system (SRS) to another. Currently, it supports conversion between geographic SRSs. For details, see Spatial Operator Functions.
SQL Syntax Notes

- **Incompatible Change**: The deprecated \texttt{ASC} or \texttt{DESC} qualifiers for \texttt{GROUP BY} clauses have been removed. Queries that previously relied on \texttt{GROUP BY} sorting may produce results that differ from previous MySQL versions. To produce a given sort order, provide an \texttt{ORDER BY} clause.

Queries and stored program definitions from MySQL 8.0.12 or lower that use \texttt{ASC} or \texttt{DESC} qualifiers for \texttt{GROUP BY} clauses should be amended. Otherwise, upgrading to MySQL 8.0.13 or higher may fail, as may replicating to MySQL 8.0.13 or higher slave servers.

XA Transaction Notes

- Previously, metadata locks for XA transactions in \texttt{PREPARED} state could be dropped when the client performing the transaction disconnected or the server restarted. This could lead to behavior such as one session being able to drop tables used by an XA transaction in progress for another session. Metadata locks for XA transactions in \texttt{PREPARED} state now are maintained across client disconnects and server restarts, until an \texttt{XA COMMIT} or \texttt{XA ROLLBACK} is executed. (Bug #22710164, Bug #79940)

X Plugin Notes

- **Important Change**: X Protocol now provides a connection pooling option, which can reduce overhead for applications that open many connections to MySQL server such as small web pages or a REST API endpoint. Use the new \texttt{mysqlx.getClient(connection, options)} operation, which returns a \texttt{Client} object. By using a \texttt{Client} object, executing an open session operation retrieves an existing and currently unused network connection from the pool, resets it, and uses it. Closing a session marks the underlying connection as unused and returns it to the \texttt{Client} object’s connection pool. The connection pool is configured using a options data dictionary, which means that a deployed application can switch from using connection pooling or not by simply changing the connection string.

  - A mutex lock has been added to prevent conflict between multiple X Plugin client sessions when a large number of sessions are being opened and closed. (Bug #28637947)

  - When an X Plugin client attempted to connect to a MySQL server specifying a database that did not exist on the server, the error message indicated that access was denied rather than that the database was not known. The correct error message is now returned. (Bug #28110957)

  - Incorrect copying of an integer value by X Plugin caused an error relating to misaligned memory access. The issue is now fixed. (Bug #28070946, Bug #90983)

  - Now that X Plugin is loaded and enabled by default, the default verbosity setting for the error log meant that no message was visible to indicate that X Plugin was available on the MySQL server. A message is now issued during system startup to confirm that X Plugin has been loaded. (Bug #27287340)

  - X Protocol now provides the ability to broadcast information to interested clients without the requirement to request the information. In addition, changes related to Group Replication are also broadcast. As part of this work the \texttt{Mysqlx_notified_by_group_replication} and \texttt{Mysqlx_notice_global_sent} status variables have been added.

Functionality Added or Changed

- **Important Change; NDB Cluster; NDB Client Programs**: Removed the deprecated \texttt{--ndb} option for \texttt{perror}. Use \texttt{ndb_perror} to obtain error message information from \texttt{NDB} error codes instead. (Bug #81705, Bug #23523957)

  References: See also: Bug #81704, Bug #23523926.
• **Important Change:** Setting user variables in statements other than `SET` is now deprecated due to issues that included those listed here:

  • The order of evaluation for expressions involving user variables was undefined.

  • The default result type of a variable is based on its type at the beginning of the statement, which could have unintended effects when a variable holding a value of one type at the beginning of a statement was assigned a new value of a different type in the same statement.

  • `HAVING`, `GROUP BY`, and `ORDER BY` clauses, when referring to a variable that was assigned a value in the select expression list, did not work as expected because the expression was evaluated on the client and so it was possible for stale column values from a previous row to be used.

  Syntax such as `SELECT @var, @var:=@var+1` is still accepted in MySQL 8.0 for backward compatibility, but is subject to removal in a future MySQL version.

• **InnoDB:** The `innodb_fsync_threshold` system variable permits a threshold size to be defined for the write buffer. By default, when InnoDB creates a new data file, such as a new log file or tablespace file, it flushes the contents of the write buffer to disk only after a file is fully written, which can cause a surge in disk write activity. The `innodb_fsync_threshold` system variable can be used to force smaller, periodic flushes to help avoid such surges in disk write activity. (Bug #27724600)

• **InnoDB:** User-created temporary tables and internal temporary tables created by the optimizer are now stored in session temporary tablespaces that are allocated to a session from a pool of temporary tablespaces. When a session disconnects, its temporary tablespaces are truncated and released back to the pool. In previous releases, temporary tables were created in the global temporary tablespace (`ibtmp1`), which did not return disk space to the operating system after temporary tables were dropped.

  The `innodb_temp_tablespaces_dir` variable defines the location where session temporary tablespaces are created. The default location is the `#innodb_temp` directory in the data directory.

  The `INNODB_SESSION_TEMP_TABLESPACES` table provides metadata about session temporary tablespaces.

  The global temporary tablespace (`ibtmp1`) now stores rollback segments for changes made to user-created temporary tables.

• **InnoDB:** The InnoDB data-at-rest encryption feature now supports general tablespaces. Previously, only file-per-table tablespaces could be encrypted. To support encryption of general tablespaces, `CREATE TABLESPACE` and `ALTER TABLESPACE` syntax was extended to include an `ENCRYPTION` clause.

  The `INFORMATION_SCHEMA.INNODB_TABLESPACES` table now includes an `ENCRYPTION` column that indicates whether or not a tablespace is encrypted.

  The `stage/innodb/alter tablespace (encryption)` Performance Schema stage instrument was added to permit monitoring of general tablespace encryption operations.

• **Replication:** You can now inspect and configure the maximum number of consensus instances at any time for a group. This maximum is referred to as the event horizon for a group, and is the maximum number of consensus instances that the system can execute in parallel. This enables you to fine tune the performance of your Group Replication deployment.

  To inspect a group's event horizon value at runtime, issue:

  ```sql
  SELECT group_replication_get_write_concurrency()
  ```
To set the maximum number of write consensus instances, issue:

```sql
SELECT group_replication_set_write_concurrency(instances);
```

where `instances` is the new number of maximum instances used for consensus.

See Using Group Replication Group Write Consensus for more information. (Bug #27260096)

**Replication:** You can now make changes to the configuration of a group while it is running, without having to stop all members to make changes. This functionality relies on UDFs which are installed with this version of the plugin, and all members of a group must have these UDFs installed. To use the UDFs, connect to an online member and issue `SELECT UDF;`

Use the `group_replication_set_as_primary()` UDF to trigger the election of a specific member as the new primary in a single-primary group, overriding the usual election process. See Changing a Group's Primary Member for more information.

In addition, you can configure the mode which a group is using while it is online, changing between single-primary mode and multi-primary mode. To change the mode of an online group, choose one of these options:

- Use `group_replication_switch_to_single_primary_mode()` to change a group running in multi-primary mode to single-primary mode.
- Use `group_replication_switch_to_multi_primary_mode()` to change a group running in single-primary mode to multi-primary mode.

See Changing a Group's Mode for more information.

**Solaris:** On Solaris, MySQL can now be built with Developer Studio 12.6. (Bug #27055190, Bug #88316, Bug #28165246, Bug #91214)

- Data truncation testing was rewritten to avoid undefined behavior. (Bug #28255956, Bug #91445)
- Out-of-range checking for float values was improved. (Bug #28225635)
- The upgrade check that the server runs during the startup process now verifies that partitioned InnoDB tables do not use shared tablespaces. (Bug #28204431)
- Previously, file I/O performed in the I/O cache in the `mysys` library was not instrumented, affecting in particular file I/O statistics reported by the Performance Schema about the binary log index file. Now, this I/O is instrumented and Performance Schema statistics are accurate. Thanks to Yura Sorokin for the contribution. (Bug #27788907, Bug #90264)
- Performance for locating user account entries in the in-memory privilege structures has been improved. Thanks to Eric Herman for the contribution. (Bug #27772506, Bug #90244)
- If `mysqld --initialize` fails to complete but creates an unusable data directory, it now displays a message that the data directory is unusable and can safely be removed. (Bug #27675647)
- Instrumentation is now provided in the Performance Schema for transaction retries by individual applier threads on a single-threaded or multithreaded slave. Previously, the Performance Schema table `replication_applier_status_by_worker` displayed information about errors that stopped the applier thread, but not about transient errors that occurred before a transaction was eventually applied. With this information, you can identify transient errors that are causing replication lag on replication slaves or Group Replication group members.
Eight new columns have been added to the Performance Schema replication_applier_status_by_worker table:

- **LAST_APPLIED_TRANSACTION_RETRIES_COUNT** - The number of times the last applied transaction was retried by the worker after the first attempt. If the transaction was applied at the first attempt, this number is zero.

- **LAST_APPLIED_TRANSACTION_LAST_TRANSIENT_ERROR_NUMBER** - The error number of the last transient error that caused the transaction to be retried.

- **LAST_APPLIED_TRANSACTION_LAST_TRANSIENT_ERROR_MESSAGE** - The message text for the last transient error that caused the transaction to be retried.

- **LAST_APPLIED_TRANSACTION_LAST_TRANSIENT_ERROR_TIMESTAMP** - The timestamp in 'YYYY-MM-DD hh:mm:ss[.fraction]' format for the last transient error that caused the transaction to be retried.

- **APPLYING_TRANSACTION_RETRIES_COUNT** - The number of times the transaction that is currently being applied was retried until this moment. If the transaction was applied at the first attempt, this number is zero.

- **APPLYING_TRANSACTION_LAST_TRANSIENT_ERROR_NUMBER** - The error number of the last transient error that caused the current transaction to be retried.

- **APPLYING_TRANSACTION_LAST_TRANSIENT_ERROR_MESSAGE** - The message text for the last transient error that caused the current transaction to be retried.

- **APPLYING_TRANSACTION_LAST_TRANSIENT_ERROR_TIMESTAMP** - The timestamp in 'YYYY-MM-DD hh:mm:ss[.fraction]' format for the last transient error that caused the current transaction to be retried.

The existing column **APPLYING_TRANSACTION_START.Apply_TIMESTAMP** is no longer reset each time a transaction is retried. It now retains the timestamp from the worker’s first attempt to apply the transaction.

- Previously, `CREATE TEMPORARY TABLE` and `DROP TEMPORARY TABLE` statements were not supported inside transactions, procedures, functions, or triggers when using GTIDs (that is, when the `enforce_gtid_consistency` system variable is set to **ON**). It was possible to use these statements with GTIDs enabled, but only outside of any transaction, and only with `autocommit=1`.

From MySQL 8.0.13, this restriction has been removed when `binlog_format` is set to **ROW** or **MIXED**. With row-based logging in use, `CREATE TEMPORARY TABLE` and `DROP TEMPORARY TABLE` statements can now be used inside transactions, procedures, functions, or triggers when GTIDs are enabled. When `binlog_format` is set to **STATEMENT**, the restriction remains. Because of this difference in behavior, some additional restrictions now apply to changing the `binlog_format` setting at runtime:

- If a session has open temporary tables, the replication format cannot be changed for the session (`SET @@SESSION.binlog_format`).

- If any replication channel has open temporary tables, the replication format cannot be changed globally (`SET @@GLOBAL.binlog_format` or `SET @@PERSIST.binlog_format`).

- If any replication channel applier thread is currently running, the replication format cannot be changed globally (`SET @@GLOBAL.binlog_format` or `SET @@PERSIST.binlog_format`).
Trying to switch the replication format in any of these cases (or attempting to set the current replication format) results in an error. You can, however, use `PERSIST_ONLY (SET @@PERSIST_ONLY.binlog_format)` to change the replication format at any time, because this action does not modify the runtime global system variable value, and takes effect only after a server restart.

When `binlog_format` is set to `ROW` or `MIXED`, `CREATE TEMPORARY TABLE` and `DROP TEMPORARY TABLE` statements are not written to the binary log and are therefore not replicated to slaves. When they are used in transactions, if the removal of these statements from the transaction results in an empty transaction, the transaction is not written to the binary log. If a transaction involving these statements is rolled back, a warning message is issued stating that the creation or dropping of the temporary tables could not be rolled back.

- The MySQL Server code for binary logging has been refactored to create new internal interfaces for accessing events in a binary log and relay log. The new interfaces separate the write and read processes for binary logging into input and output streams, and decouple the process of capturing and retrieving binary log events from the process of writing them to files. A logical binary log file is used to wrapper the storage layer operations.

The new internal interfaces make it possible for MySQL Server to use alternative storage methods for binary log events besides a standard binary log or relay log file, including a binary log cache or memory buffer. For example, Group Replication uses the new interfaces to serialize events directly to memory buffers and transaction messages in order to co-ordinate transactions in the group. `mysqlbinlog` also now uses them to read binary log events from standard input.

The following existing error messages are now marked as obsolete due to the new internal interfaces:

- `ER_BINLOG_CANT_OPEN_LOG`
- `ER_BINLOG_CANT_CREATE_CACHE_FOR_LOG`
- `ER_BINLOG_ERROR_GETTING_NEXT_LOG_FROM_INDEX`
- `ER_RPL_RECOVERY_ERROR_FREEING_IO_CACHE`
- `ER_GRP_RPL_REINIT_OF_INTERNAL_CACHE_FOR_READ FAILED`
- `ER_GRP_RPL_APPENDING_DATA_TO_INTERNAL_CACHE_FAILED`
- `ER_GRP_RPL_REINIT_OF_INTERNAL_CACHE_FOR_WRITE_FAILED`
- `ER_GRP_RPL_FAILED_TO_CREATE_COMMIT_CACHE`
- `ER_GRP_RPL_REINIT_OF_COMMIT_CACHE_FOR_WRITE_FAILED`

- The `TempTable` storage engine now supports storage of binary large object (BLOB) type columns. This enhancement improves performance for queries that use temporary tables containing BLOB data. Previously, temporary tables that contained BLOB data were stored in the on-disk storage engine defined by `internal_tmp_disk_storage_engine`.

- The MySQL Server code for deserialization of binary log events has been refactored to improve the identification and resilient handling of corrupted and invalid event data during replication, and also during processing with `mysqlbinlog`, whether or not binary log checksums are used on the server.

- Previously, executing `RENAME TABLE` required that there be no tables locked with `LOCK TABLES`. Now it is possible to rename tables that are locked with a `WRITE` lock or that are the product of renaming `WRITE`-locked tables from earlier steps in a multiple-table rename operation.
• The `group_replication_member_expel_timeout` option has been added to define the time period that the group should wait for a non-responding member before evicting the member from the group. This enables you to configure the eviction process when the connection to members is unreliable.

**Bugs Fixed**

• **Important Note:** The `libevent` library included with the MySQL Server was upgraded to version 2.1.8. (Bug #28207237, Bug #29041505, Bug #29055011)

• **InnoDB; Partitioning:** Removed old `InnoDB` handler and partitioning code that referenced `.frm` files, and thus no longer had any purpose. (Bug #27995316)

• **InnoDB:** An assertion was raised during a DROP TABLE operation. A thread that was accessing the table through the memcached API released metadata locks before releasing the table. (Bug #28531148)

• **InnoDB:** The `being_modified` bit in a LOB reference was set but the bit modification was not logged, causing an assertion failure. (Bug #28443837)

• **InnoDB:** Window functions returned incorrect results when the optimizer used the `InnoDB` storage engine for internal temporary tables. (Bug #28430650)

• **InnoDB:** Adjusting the server time to an earlier time caused periodic redo flushes to be missed. (Bug #28430358, Bug #90670)

• **InnoDB:** An ALTER TABLE operation that added a primary key produced a segmentation fault. (Bug #28395278)

  References: This issue is a regression of: Bug #27753193.

• **InnoDB:** A conditional check was removed by removing the `ReadView::complete()` function and splitting its work among other functions. This change helps optimize performance on ARM 64-bit. (Bug #28385211, Bug #91759)

• **InnoDB:** Leftover `thread_mutex` code was removed from `InnoDB` source code files. (Bug #28363673, Bug #91678)

• **InnoDB:** Type changes were implemented to eliminate warnings that occurred when compiling `InnoDB` with Microsoft Visual Studio 2017. (Bug #28338720)

• **InnoDB:** An invalid assertion was raised when a B-tree flag used to mark shared index locks was used to mark a shared-exclusive index lock. (Bug #28317172)

• **InnoDB:** The sharp checkpoint mechanism no longer forces preflushing of dirty pages when requesting a checkpoint for the currently available LSN.

  The log checkpoint thread now takes the concurrency margin (the per thread margin for free space in the log) into account when determining if the next checkpoint write is required and whether to wake up page cleaners to force a sync-flush of dirty pages. Page cleaner threads take the concurrency margin into account when determining whether to flush dirty pages and how many pages to flush. (Bug #28297462)

• **InnoDB:** A misplaced debug crash point caused a transaction timeout resulting in test failures. (Bug #28295814)

• **InnoDB:** `InnoDB` error message format was modified to remove duplicate text. (Bug #28289789)

• **InnoDB:** Unnecessary cycles of freeing and allocating memory caused JSON performance degradation on Windows. (Bug #28278737)
• **InnoDB:** InnoDB incorrectly permitted dropping an index used in a foreign key constraint using an in-place `ALTER TABLE` statement. This operation is now blocked at the SQL layer. (Bug #28268875)

• **InnoDB:** To avoid checking hardware support each time a hardware-optimized checksum is computed, asserts were converted to debug-only asserts. (Bug #28267334, Bug #91485)

• **InnoDB:** A patch that combined Variance-Aware Transaction Scheduling (VATS) with functionality that releases read locks caused gap locks to be removed without granting locks to waiting transactions, resulting in transaction timeouts. (Bug #28261530)

  References: This issue is a regression of: Bug #28261530.

• **InnoDB:** The `log_checkpoint` thread failed to write new checkpoints in a timely manner when the amount of redo was small. (Bug #28220222)

• **InnoDB:** The server exited during an in-place upgrade from MySQL 5.7 to MySQL 8.0 due to an attempted eviction of a foreign-key-related table from the cache. At the end of the upgrade process, tables with `FULLTEXT` indexes were marked as ready for eviction without checking for foreign key relationships. (Bug #28212734, Bug #91325)

• **InnoDB:** The format of the following Performance Schema and `INFORMATION_SCHEMA` table columns was modified:

  - `data_locks.ENGINE_LOCK_ID`
  - `data_lock_waits.REQUESTING_ENGINE_LOCK_ID`
  - `data_lock_waits.BLOCKING_ENGINE_LOCK_ID`
  - `INNODB_TRX.TRX_REQUESTED_LOCK_ID`

  The previous format was `trx_id:table_id` for table locks and `trx_id:space_id:page_no:heap_no` for record locks. The new format is `trx_immutable_id:table_id:lock_immutable_id` for table locks and `trx_immutable_id:space_id:page_no:heap_no:lock_immutable_id` for record locks.

  `lock_immutable_id` and `trx_immutable_id` are 64-bit values that do not change during the lifetime of a lock or transaction, respectively, and are unique among other instance object IDs. (Bug #28176910)

• **InnoDB:** The list of permitted lock mode descriptors used by the `LOCK_MODE` column of the Performance Schema `data_locks` table was expanded to include `REC_NOT_GAP`, `INSERT_INTENTION`, `PREDICATE`, and `PRDT_PAGE`. `REC_NOT_GAP` indicates a record-only lock. `INSERT_INTENTION` indicates an insert intention lock. `PREDICATE` and `PRDT_PAGE` descriptors indicate a spatial index lock. (Bug #28176805)

• **InnoDB:** Table names were not compared in lowercase on macOS with a setting of `lower_case_table_names=2`, which caused instability after restarting the server. (Bug #28170699, Bug #91204)

• **InnoDB:** Macros used to define constant values in InnoDB source code were changed to constant expressions. (Bug #28152926)

• **InnoDB:** A flag that prevents transactions from being rolled back during commit is now set earlier to prevent scenarios in which a high priority transaction could abort a transaction that is in the process of being committed. (Bug #28140462)
- **InnoDB**: A query that scanned the primary key of a table did not return the expected result. (Bug #28104394, Bug #91032)

- **InnoDB**: Unnecessary header file inclusions were removed from InnoDB source code files. (Bug #28086759)

- **InnoDB**: An apparent hang due to the log writer running out of free space in the redo log caused the server to exit. (Bug #28072385, Bug #90993)

- **InnoDB**: A query interruption during a lock wait caused an error. (Bug #28068293)

- **InnoDB**: After upgrading from MySQL 5.7 to MySQL 8.0, invalid warnings indicated that undo tablespace IDs were not in the undo tablespace range. The warnings occurred if the MySQL 5.7 installation was configured to use separate undo tablespaces. (Bug #28060337)

- **InnoDB**: Error messaging was improved for startup failures on an incomplete cloned data directory. (Bug #28032131)

- **InnoDB**: A segmentation fault occurred during an `XA COMMIT` operation. (Bug #27995891)

- **InnoDB**: Unused code was removed from the TempTable storage engine source code. (Bug #27978968)

- **InnoDB**: An index record was not found when updating a secondary index defined on a generated column. (Bug #27968952)

- **InnoDB**: A spurious negation operator in an IF condition caused the Variance-Aware Transaction Scheduling (VATS) algorithm to be used for table locks. (Bug #27944920)

  References: This issue is a regression of: Bug #27572937.

- **InnoDB**: The update log applied as part of an online `ALTER TABLE` operation did not take into account the computed value of the generated column in the old row while updating the secondary index. (Bug #27921932)

- **InnoDB**: Buffer pool memory allocation was not fully accounted for in Performance Schema memory/innodb/buf_buf_pool statistics. (Bug #27917595, Bug #90561)

- **InnoDB**: An unsupported DDL operation involving a foreign key constraint raised an assertion. (Bug #27912873)

- **InnoDB**: A lock-related debug assertion failure was raised when more than one lock matched a search condition, which could lead to releasing the wrong lock. (Bug #27898384)

- **InnoDB**: A function that removes aborted indexes during the prepare phase of an online `ALTER TABLE` operation did not record its changes. (Bug #27879325)

- **InnoDB**: A general tablespace created in MySQL 5.7 with no assigned table caused a failure when upgrading to MySQL 8.0. (Bug #27877485)

- **InnoDB**: Concurrent undo tablespace truncation and master key rotation operations raised an assertion. (Bug #27872369)

- **InnoDB**: A boolean marker identifying whether a transaction holds a mutex was not placed in the correct location. (Bug #27870035)

- **InnoDB**: An attempted foreign key check on a discarded table caused a segmentation fault. (Bug #27804668)
MySQL 8.0 Release Notes

- **InnoDB**: B-tree bulk load operations could leave a page in a partially initialized state. (Bug #27802098)

- **InnoDB**: Starting the server inside a Docker container on a NUMA enabled operating system raised an “mbind: Operation not permitted” error. (Bug #27792853)

- **InnoDB**: A partitioned table `TABLE_ID` value stored in a storage-engine-private data field in the data dictionary was not adjusted properly after an `ALTER TABLE ... PARTITION` operation. (Bug #27784462)

- **InnoDB**: The server halted with a “log writer overwriting data after checkpoint - waited too long” error. (Bug #27779266)

- **InnoDB**: With `innodb_flush_log_at_trx_commit=2`, the log flusher thread could wait for an event with a timeout period equal to the `innodb_flush_log_at_timeout` setting, causing an initialization delay. (Bug #27762596)

- **InnoDB**: An assertion was raised during an `OPTIMIZE TABLE` operation. (Bug #27753193)

- **InnoDB**: Transaction rollback due to a deadlock caused an assertion failure in debug builds. Initiation of an attachable transaction for accessing the data dictionary was not expected during transaction rollback. (Bug #27729974)

- **InnoDB**: With `innodb_flush_log_at_trx_commit=0` and binary logging enabled, redo logs were not flushed as expected during the commit phase of DDL operations. (Bug #27691035)

- **InnoDB**: With `REDDUNDANT` or `COMPRESSED` row format and `READ COMMITTED` isolation level, only the LOB value prefix and possibly the external part of an old LOB value were returned, which could cause a JSON document to be viewed as corrupted. If there was no LOB value prefix, an old LOB value with new values for other fields could be returned, causing data inconsistency. (Bug #27624990)

- **InnoDB**: A debug option that permits pausing periodic checkpoints became obsolete after periodic checkpoint ownership was moved from the master thread to a log checkpointer thread. The debug option was replaced by another method of pausing periodic checkpoints. (Bug #27588328)

- **InnoDB**: A transaction on a table with a spatial index defined on a column with a spatial reference identifier (SRID) was able to insert into an area selected for update by another transaction. (Bug #27577612)

- **InnoDB**: A foreign key constraint name was duplicated during a rename table operation, causing a failure during later query execution. (Bug #27545888)

- **InnoDB**: A serialized dictionary information (SDI) deletion failure raised an assertion. (Bug #27493634)

- **InnoDB**: A server exit occurred after freeing large object (LOB) index entries during a LOB purge or rollback. (Bug #27419474)

- **InnoDB**: In a function called before the execution of a statement in a stored procedure, a read and write operation on `trx->lock.start_stmt` was not protected by a mutex. (Bug #27325898)

- **InnoDB**: The `INFORMATION_SCHEMA.FILES` and `INFORMATION_SCHEMA.INNODB_TABLESPACES` tables did not show the actual undo tablespaces that were present in the MySQL instance. Only the two default undo tablespaces were shown. (Bug #26820406)

- **InnoDB**: An error occurred during a DDL operation due to a mismatch in a `REDDUNDANT` row format calculation that determines the length of the online log. (Bug #26375771)

- **InnoDB**: Enabling `innodb_undo_log_truncate` negatively affected transaction processing performance. Instead of performing two checkpoints during an undo tablespace truncate operation, pages that belong to the tablespace file are now flushed from disk. (Bug #26322656)
• **InnoDB:** A helper class was introduced to improve performance associated with reading from secondary keys when there are multiple versions of the same row.

Thanks to Domas Mituzas for the contribution. (Bug #25540277, Bug #84958)

• **InnoDB:** The location of the Innodb Merge Temp File that reported by the `wait/io/file/innodb/innodb_temp_file` Performance Schema instrument was incorrect. (Bug #21339079, Bug #77519)

• **Partitioning:** When a `CREATE TABLE ... PARTITION BY ...` statement failed due to an invalid partition definition, the server did not remove any partition files which might have been created prior to encountering the invalid `PARTITION` clause. (Bug #27798708)

References: See also: Bug #88043, Bug #26945644.

• **Partitioning:** It was possible to perform `FLUSH TABLES FOR EXPORT` on a partitioned table created with `innodb_file_per_table=1` after discarding its tablespace. Attempting to do so now raises `ER_TABLESPACE_DISCARDED`. (Bug #90545, Bug #27903881)

References: See also: Bug #80669, Bug #22899690.

• **Partitioning:** An extraneous row lock was imposed by an update to a partitioned InnoDB table. (Bug #87253, Bug #26553164)

• **Replication:** When a replication slave is restarted by a `START SLAVE` statement, the columns in the Performance Schema table `replication_applier_status_by_worker` beginning `APPLYING_TRANSACTION` are now reset on a slave that is operating in single-threaded mode. These columns were always reset on a multi-threaded slave, because the existing worker threads were terminated by the statement and the information could not be retained. The behavior has now been standardized across the slave configurations by resetting the columns for a single-threaded slave as well. (Bug #28248026)

• **Replication:** If a multi-threaded replication slave was stopped, changed to a single-threaded slave (by setting `slave_parallel_workers > 0`), and restarted, the Performance Schema table `replication_applier_status_by_worker` showed irrelevant timestamps because the old monitoring information had not been cleared. (Bug #28191382)

• **Replication:** When the `binlog_group_commit_sync_delay` system variable is set to a wait time to delay synchronization of transactions to disk, and the `binlog_group_commit_sync_no_delay_count` system variable is also set to a number of transactions, the MySQL server exits the wait procedure if the specified number of transactions is reached before the specified wait time is reached. The server manages this process by checking on the transaction count after a delta of one tenth of the time specified by `binlog_group_commit_sync_delay` has elapsed, then subtracting that interval from the remaining wait time.

If rounding during calculation of the delta meant that the wait time was not a multiple of the delta, the final subtraction of the delta from the remaining wait time would cause the value to be negative, and therefore to wrap to the maximum wait time, making the commit hang. The data type for the remaining wait time has now been changed so that the value does not wrap in this situation, and the commit can proceed when the original wait time has elapsed. Thanks to Yan Huang for the contribution. (Bug #28091735, Bug #91055)

• **Replication:** In debug builds, an assertion failed because more than 255 collations are now available in MySQL. (Bug #28015761)

• **Replication:** An assertion was raised in debug builds because the MySQL server recorded a GTID consistency violation, but did not remove the record after the relevant statement failed to execute
MySQL 8.0 Release Notes

successfully. The handling of this situation has now been improved to ensure that the server checks at the end of a transaction whether a GTID consistency violation was produced by a failed statement, and if this is the case, restores the previous GTID consistency state. (Bug #27903831, Bug #90551)

- **Replication:** With GTIDs in use for replication, transactions including statements that caused a parsing error (ER_PARSE_ERROR) could not be skipped manually by the recommended method of injecting an empty or replacement transaction with the same GTID. This action should result in the slave identifying the GTID as already used, and therefore skipping the unwanted transaction that shared its GTID. However, in the case of a parsing error, because the statement was parsed before the GTID was checked to see if it needed to be skipped, the replication applier thread stopped due to the parsing error, even though the intention was for the transaction to be skipped anyway.

  With this fix, the replication applier thread now ignores parsing errors if the transaction concerned needs to be skipped because the GTID was already used. Note that this behavior change does not apply in the case of workloads consisting of binary log output produced by mysqlbinlog. In that situation, there would be a risk that a transaction with a parsing error that immediately follows a skipped transaction would also be silently skipped, when it ought to raise an error. (Bug #27638268)

- **Replication:** When a **RESET SLAVE** statement was issued on a replication slave with GTIDs in use, the existing relay log files were purged, but the replacement new relay log file was generated before the set of received GTIDs for the channel had been cleared. The former GTID set was therefore written to the new relay log file as the **PREVIOUS_GTIDS** event, causing a fatal error in replication stating that the slave had more GTIDs than the master, even though the **gtid_executed** set for both servers was empty. Now, when **RESET SLAVE** is issued, the set of received GTIDs is cleared before the new relay log file is generated, so that this situation does not occur. (Bug #27636289)

- **Replication:** The master's receiver thread for semisynchronous replication held a mutex while reading acknowledgements from slaves, but the same mutex was required to add or remove a semisynchronous slave, causing those operations to be delayed by the acknowledgement activity. The issue has now been fixed by not acquiring the mutex to read the acknowledgements from slaves. (Bug #27610678, Bug #89370)

- **Replication:** In code for replication slave reporting, a rare error situation raised an assertion in debug builds, but in release builds, returned leaving a mutex locked. The mutex is now unlocked before returning in this situation. Thanks to Zsolt Parragi for the patch. (Bug #27448019, Bug #89421)

- **Replication:** Entries in the relay log info log (the **slave_relay_log_info** table) for the Group Replication-specific channels **group_replication_applier** and **group_replication_recovery** were not being cleared by a **RESET SLAVE** or **RESET SLAVE ALL** command. (Bug #27411175)

- **Replication:** Automatic retrying of transactions on a replication slave, as specified by the **slave_transaction_retries** system variable, was taking place even if the transaction had a non-temporary error that would repeat on retrying or that indicated wider issues. Now, transactions are only automatically retried if there is either no error, or an error that is only temporary. (Bug #27373559, Bug #89143)

- **Replication:** When **FLUSH** statements for specific log types (such as **FLUSH SLOW LOGS**) resulted in an error, the statements were still written to the binary log. This stopped replication because the error had occurred on the master, but did not occur on the slave. MySQL Server now checks on the outcome of these **FLUSH** statements, and if an error occurred, the statement is not written to the binary log. (Bug #24786290, Bug #83232)

- **Replication:** The **PASSWORD()** function, which produces a hash of the password, was deprecated in MySQL 5.7 and removed in MySQL 8.0. When a **SET PASSWORD** statement that used this function was replicated from a MySQL 5.6 master to a MySQL 5.7 slave, or from a MySQL 5.7 master with the **log_builtin_as_identified_by_password** system variable set to ON to a MySQL 5.7 slave, the
password hash was itself also hashed before being stored on the slave. The issue has now been fixed and the replicated password hash is stored as originally passed to the slave. (Bug #24687073)

- **Replication:** If an `ORDER BY` clause was used in retrieving records from certain Performance Schema tables relating to replication, an empty set was returned. The issue has now been fixed. (Bug #22958077, Bug #80777)

- **Replication:** When replication channels are used on a slave for multi-source replication, a `START SLAVE` statement that does not specify an individual channel (so without the `FOR CHANNEL` clause) should start the I/O thread and the SQL thread for all of the channels on the replication slave. However, if a `RESET SLAVE` statement was used on such a slave, a subsequent `START SLAVE` statement did not start the non-default channels. Now, replication channels that are deinitialized as a result of a `RESET SLAVE` statement, rather than as a result of an error in the initialization process, are identified and are restarted by a `START SLAVE` statement that applies to all channels. (Bug #22809607)

- **Replication:** Issuing `RESET SLAVE` on a replication slave does not change any replication connection parameters such as master host, master port, master user, or master password, which are retained in memory. However, these connection parameters are reset if you issue `RESET SLAVE ALL`. Previously, if the slave `mysqld` was restarted immediately after issuing `RESET SLAVE` (including a server crash as well as a deliberate restart), the connection parameters were reset as if `RESET SLAVE ALL` had been used.

Now, when `master_info_repository=TABLE` is set on the server (which is the default from MySQL 8.0), replication connection parameters are preserved in the crash-safe InnoDB table `mysql.slave_master_info` as part of the `RESET SLAVE` operation. They are also retained in memory. In the event of a server crash or deliberate restart after issuing `RESET SLAVE` but before issuing `START SLAVE`, the replication connection parameters are retrieved from the table and reused for the new connection.

If `master_info_repository=FILE` is set on the server (which is the default in MySQL 5.7), replication connection parameters are only retained in memory, so the behavior remains the same as previously. If the slave `mysqld` is restarted due to a server crash or a deliberate restart immediately after issuing `RESET SLAVE`, the connection parameters are lost. In that case, you must issue a `CHANGE MASTER TO` statement after the server start to respecify the connection parameters before issuing `START SLAVE`.

If you want to reset the connection parameters intentionally, you need to use `RESET SLAVE ALL`, which clears the connection parameters. In that case, you must issue a `CHANGE MASTER TO` statement after the server start to specify the new connection parameters. (Bug #20280946)

- **Replication:** When a group member resumes after being suspended for some time and is not able to process all pending messages, it enters the `ERROR` state. However, the remaining members see it as `UNREACHABLE`, and wait until the member's suspicion expires to evict it from the group. The behavior has now been modified and a member stopping due to some error tries to connect to a known peer to request its removal from the group, before installing the leave view. (Bug #91433, Bug #28252687)

- **Replication:** If Group Replication was started on server with an invalid `group_replication_group_name`, the server would stop unexpectedly. (Bug #91347, Bug #28219136)

- **Replication:** Compilation warnings related to unused functions in `xdr_utils` have been reduced. Thanks to Zsolt Parragi for the patch. (Bug #91071, Bug #28099963)

- **Replication:** If the `group_replication_recovery_retry_count` variable was modified while the member was already making a reconnection attempt, the connection attempt could enter an infinite loop. (Bug #91057, Bug #28092714)
• **Replication:** The `group_replication_exit_state_action` variable enables you to specify what action is taken if a member involuntarily leaves the group, but when starting a server with `group_replication_start_on_boot` enabled the `group_replication_exit_state_action` variable was being ignored during the following scenarios:

  • valid number of group members was exceeded
  • incompatible configuration of the member system variables (various)
  • the joining member had more transactions than the group
  • the joining member’s version was not compatible with the group
  
  (Bug #90494, Bug #27881311)

• **Replication:** When `group_replication_group_seeds` contained a DNS based entry which resolved to its own local address, Group Replication could not start. (Bug #90483, Bug #27882096, Bug #28074929)

• **Replication:** Issuing `START GROUP_REPLICATION` and then forcibly stopping the `mysqld` process, for example using `control-C`, could result in an unexpected halt of the server. (Bug #90457, Bug #27873419)

• **Microsoft Windows:** An error now is written to the server log when the presence of the `NO_AUTO_CREATE_USER` value for the `sql_mode` option in the options file prevents a MySQL 8.0 server from starting. (Bug #28061945, Bug #90967)

• **Microsoft Windows:** On Windows, uninstallation of the MySQL Server MSI package through MySQL Installer produced a spurious popup window. (Bug #27463864)

• **Microsoft Windows:** On Windows, `DBUG_ABORT` did not print the custom stack trace and other information. (Bug #21383530)

• **Microsoft Windows:** Starting MySQL as a Windows service with the service-installation command that specified a service name following the `--install` option ignored the directives in the named service group of the `my.ini` or `my.cnf` options files and used default options instead. Only the default service names (`mysqld`, `mysql_cluster`, `server`, `mysqld-8.0`) could load different parameters from an options file. (Bug #90383, Bug #27852209)

• **JSON:** The server did not reject creation of a table with a generated column in which the generated column used `JSON_TABLE()`, even though subqueries, parameters, variables, stored functions, and user-defined functions are not permitted in expressions for generated columns. The server now checks more aggressively to make sure any of the disallowed constructs (including `JSON_TABLE()`) are rejected for use in such expressions. (Bug #28518485)

• **JSON:** `SELECT ... FROM JSON_TABLE()` sometimes failed with a permissions error for a user other than MySQL root. This issue could also occur when such a query was used as the basis for a view, and a `SELECT` from the view failed. (Bug #28255453, Bug #27923406)

  References: See also: Bug #27189940.

• **JSON:** The `JSON_TABLE()` function subjected integer values greater than or equal to $2^{31}$ to wraparound. For example, the query `SELECT id FROM JSON_TABLE('[["id":2147483648"]]', '[$[*]]' COLUMNS (id BIGINT UNSIGNED PATH '.$id')) AS json` returned `-2147483648`. (Bug #27856835)

• **JSON:** In some contexts, the `NULLIF()` function returned its first argument as a boolean value rather than its actual type. This was noticed when the result of this function was used as an argument
to `JSON_ARRAYAGG()` or `JSON_OBJECTAGG()`, but could have occurred in other cases in which `NULLIF()` was used in a similar fashion. (Bug #90833, Bug #28007237)

- **JSON:** When a JSON document which contained binary data was converted to base-64 encoded text for display, newline characters in the encoded string were not properly escaped, so that the text representation could not be parsed as JSON, and was thus truncated, corrupted, or both. Now MySQL makes sure that any newline characters in the encoded string are escaped. (Bug #90503, Bug #27891359)

- **filesort** operations could cause a server exit. (Bug #28791531, Bug #92777)

- Debian packaging now supports Ubuntu 18.10 (Cosmic Cuttlefish). (Bug #28765706)

- On the Fedora 29 platform, OpenSSL 1.0.x is used to build packages because 1.1.1 support is not ready. If you build MySQL from source, it is recommended that you build using the `compat-openssl10-devel` package. (Bug #28737143)

- On the Fedora 29 platform, upgrading from MariaDB to MySQL 8.0.13 failed due to missing obsoletes. (Bug #28727698)

- MySQL binary distributions for SLES 12 now are built using GCC 7. The lowest supported GCC version on this platform is now 5.3 (previously 4.8.5). (Bug #28542723)

- Trying to add a functional index on a subquery should not be possible, and caused the server to hit an assertion when trying to resolve the indexed expression. Now in such cases, the expression is disallowed, and the servers returns an appropriate error message. (Bug #28526493)

- Added a range check when performing calculations with exponents. (Bug #28505423)

- `CMAKE -DWITHOUT_SERVER=1` resulted in build errors. (Bug #28501563, Bug #92011)

- When a prefix index was specified with a length of 8 bytes, the `Sub_part` column in the output of `SHOW INDEXES` was `NULL`. (Bug #28499603)

- Compilation failed for GCC 8 with MySQL configured to use some system libraries. (Bug #28471072, Bug #91914)

- For debug builds, if the server bootstrapping phase failed, missing cleanup code caused an assertion to be raised. (Bug #28435378, Bug #91847)

- Numeric ranges in MySQL builds could differ between the ARM and Intel x64 platforms. (Bug #28401869)

- For `mysqldump --tables` output, file names now always include a `.txt` or `.sql` suffix, even for file names that already contain a dot. Thanks to Facebook for the contribution. (Bug #28380961, Bug #91745)

- Concurrent `INSERT` and `SELECT` statements on a `MERGE` table could result in a server exit. (Bug #28379285)

- On SLES 15, upgrading from MariaDB packages to MySQL packages failed due to incorrect `obsoletes` information. (Bug #28292138)

- The data dictionary auto releaser now allocates maps for data dictionary object types dynamically. Also, maps were added for column statistics and resource groups used in size calculations and object removal, and an auto releaser was added for `ANALYZE TABLE` operations. (Bug #28245522, Bug #91420)
MySQL 8.0 Release Notes

- It was possible for a `UNION` query that mixed different character sets to fail with a spurious error. (Bug #28237675)
  
  References: This issue is a regression of: Bug #83895, Bug #25123839.

- `CHECK TABLE ... FOR UPGRADE` on temporary tables could raise an assertion. (Bug #28220374)
  
  References: This issue is a regression of: Bug #24741307.

- A divide-by-zero error could occur in the range optimizer. (Bug #28214186)

- Due to a GCC 8 bug, recursion in foreign key checks could exhaust stack space. (Bug #28200774, Bug #28421040, Bug #91823)

- When `sql_mode` was set to the empty string, queries of the form
  
  ```
  SELECT CONCAT( FORMAT(LPAD(char,2,''), 1) )
  ```

  were not handled properly; the same was also true for similarly-formed queries using `RPAD()` in place of `LPAD()`. (Bug #28197977)

- Non-privileged users could change their own account password history and reuse properties. (Bug #28191838)

- The OpenSSL 1.1 `OPENSSL_malloc_init()` call did not work well on Windows. To address this, alternative allocation wrappers are used instead of the defaults from OpenSSL. Consequently, OpenSSL memory instrumentation is turned off. (Bug #28179051)

- Allocation of certain object types during data dictionary lookups resulted in a stack buffer overflow. (Bug #28176453)

- An internal server operation that attempted to perform a commit while fetching table statistics from the `INFORMATION_SCHEMA` could raise an assertion. (Bug #28165060)

- Certain stored procedures could cause a server exit. (Bug #28156802)

- Output for `CREATE USER` statements could differ in the general query log and audit log. (Bug #28147710)

- For `UPDATE` and `DELETE` statements that produce an error due to `sql_safe_updates` being enabled, the error message was insufficiently informative. The message now includes the first diagnostic that was produced, to provide information about the reason for failure. For example, the message may indicate that the `range_optimizer_max_mem_size` value was exceeded or type conversion occurred, either of which can preclude use of an index.

  Additionally: (1) Using `EXPLAIN` for such statements does not produce an error, enabling users to see from `EXPLAIN` plus `SHOW WARNINGS` output why an index is not used. (2) For multiple-table deletes and updates, an error is produced with safe updates enabled only if any target table uses a table scan. (Bug #28145710, Bug #91080)

- MySQL Server and test RPM packages were missing `perl-Data-Dumper` as a dependency. (Bug #28144933, Bug #72926)

- Server startup could fail if a service needed by a component was not yet initialized. (Bug #28142250)

- `SHOW CREATE TABLE` could omit foreign key `RESTRICT` options. This in turn could cause foreign key `RESTRICT` options to be lost from tables dumped with `mysqldump` and restored from the dump file. (Bug #28122781, Bug #91110)

- The `mysql` client was slow to import large multiple-line statements in batch mode. Memory allocation is now more efficient for this situation. Thanks to Sinisa Milivojevic for the patch. (Bug #28116512, Bug #85155)
• The **SUM()** and **AVG()** functions did not handle correctly a string argument used with a window function. (Bug #28105241)

• Excessive nesting of geometry collections caused the server to exhaust stack space. The server now raises an error if there is a danger of stack overrun. (Bug #28100563)

• For the **mysql** client, the --b short option was associated with two long options, --no-beep and --binary-as-hex. The --b option now is associated only with --no-beep. (Bug #28093271)

• For a table with an auto-increment primary key, concurrent ALTER TABLE ... ADD ... VIRTUAL and INSERT statements could lead to duplicate-key errors. (Bug #28089240)

• Handling of floating-point values by **SUM()** was improved. (Bug #28080199)

• Some expressions using **ST_Simplify()** could cause a server exit. (Bug #28079969)

• In builds with Undefined Behavior Sanitizer enabled, testing returned a “member call on null pointer of type ’struct Event_db_repository” error. (Bug #28066155)

• The **WITH_GMOCK** CMake option did not handle Windows path names properly. (Bug #28061409, Bug #90964)

• **INFORMATION_SCHEMA** queries that attempted to cache table statistics could raise a debugging assertion. (Bug #28035207)

• If roles were assigned to accounts, certain **sql_mode** settings could cause the server to be unable to start. (Bug #28030423)

• Group lookups for LDAP authentication plugins could fail if the user had insufficient privileges. Now, group search operations bind again using **root** credentials if those are available. (Bug #28016008)

• **ANALYZE TABLE ... UPDATE HISTOGRAM** statements produced by **mysqldump** contained a syntax error. (Bug #28014376, Bug #90846)

• Generated columns having indexes and that used a string function were not always populated correctly. (Bug #27973409)

• Fixed-length sort keys, such as those used by priority queues, that fit exactly were assumed to have failed. (Bug #27970481, Bug #92448, Bug #28654343)

• For an empty result, **REPLACE()** sometimes returned a null string rather than an empty string. (Bug #27960921)

• Dropping a table that was created with a user-defined collation no longer available could cause a server exit. (Bug #27952999)

• Joining the **INFORMATION_SCHEMA REFERENTIAL_CONSTRAINTS** and **TABLE_CONSTRAINTS** tables on the **CONSTRAINT_NAME** failed because the column collations differed. (Bug #27945704, Bug #90690)

• Some operations on **DECIMAL** values could cause a server exit. (Bug #27942277)

• Stored program definitions in **mysqldump** dump files sometimes included the **NO_AUTO_CREATE_USER** **SQL mode.** Because that mode has been removed in MySQL 8.0, loading such a dump file into a MySQL 8.0 server failed. **mysqldump** now removes **NO_AUTO_CREATE_USER** from the definition of dumped stored programs. (Bug #27931181, Bug #90624)

• Very long table keys were handled incorrectly on replication slaves. (Bug #27930505)

• **mysqld** did not determine its installation directory correctly if that directory was the last one listed in $PATH. (Bug #27922896)
• During server startup/shutdown, PID files could be mishandled. (Bug #27919254)

• When the server was started with `--skip-grant-tables`, some account-management statements were not disabled. (Bug #27906226)

• On rare occasions, setting a savepoint could raise an assertion. (Bug #27898591)

• Runtime errors could occur for calls to `copy_integer()`, `myfunc_int()`, `mysql_sys_var_int()`, or `thd_killed()`; for calls to MyISAM sorting functions; or for values outside the range of representable values of type `unsigned long long`. (Bug #27894901, Bug #90515, Bug #27871951, Bug #27918095, Bug #90609, Bug #27937522, Bug #90661, Bug #27978325, Bug #27962900)

• `mysql_install_plugin` failed to report plugin-specific errors if the server was unable to store plugin metadata in the data dictionary. (Bug #27893406)

• `ssl_fips_mode` cannot be set to `ON` unless the OpenSSL requirements are satisfied, but a failed attempt still resulted in the value being displayed as `ON`. (Bug #27891890)

• The symbol for the `mysql_result_metadata()` C API function was not exported by the client library on Windows. (Bug #27868095)

• For InnoDB tables, self-referencing foreign keys could be created, causing server misbehavior. (Bug #27864515)

• A `UNION` query mixing different character sets produced invalid output when performed in a prepared statement or a stored procedure. (Bug #27849293)

• Debian packages were missing a dependency for `libcurl-dev`. (Bug #27844465)

• `--help` output for client programs did not include the current `--ssl-fips-mode` value. (Bug #27838966)

• A runtime error could occur for calls to `Derived_key_comp()`. (Bug #27830679)

• For MEMORY tables, memory overflow errors could occur. (Bug #27799513)

• When converting from a `BLOB` (or `TEXT`) type to a smaller `BLOB` (or `TEXT`) type, no warning or error was reported informing about the truncation or data loss. Now an appropriate error is issued in strict SQL mode and a warning in nonstrict SQL mode. (Bug #27788685, Bug #90266)

• For debug builds, `my_strnxfrm_unicode_full_bin()` could raise an assertion. (Bug #27752619)

• In builds with Undefined Behavior Sanitizer enabled, signed integer overflow could occur in GIS calculations. (Bug #27751479, Bug #27744399, Bug #27811282)

• It was possible in some cases for the internal function `regexp::EvalExprToCharset()` to bind a reference to a misaligned address. (Bug #27743722)

• The severity of messages produced by the server about being unable to read key files has been escalated from `INFORMATION` to `WARNING`. (Bug #27737195)

• No temporary table field was created for an expression using window functions combined with a rollup, causing queries containing these to fail. (Bug #27735167)

• The expression `FIND_IN_SET( JSON_UNQUOTE( JSON_SET('{}','$','') ), 1)` was not evaluated properly. (Bug #27731699)

• Failure to create a temporary table during a MyISAM query could cause a server exit. Thanks to Facebook for the patch. (Bug #27724519, Bug #90145)
MySQL 8.0 Release Notes

• parser_max_mem_size was ineffective when parsing stored program definitions. (Bug #27714748)
• A BETWEEN clause comparing negative values could lead to erroneous results. (Bug #27691347)
• Some typos in server error messages were fixed. Thanks to Thomas Tsiakalakis for the contribution. (Bug #27688294, Bug #90048)
• These scripts are no longer included in RPM packages (they are unnecessary because they are compiled into the mysqld binary): fill_help_tables.sql, mysql_sys_schema.sql, mysql_system_tables.sql, mysql_system_tables_data.sql, mysql_system_users.sql. (Bug #27672991)
• Unique::io_cache was not cleaned up properly when restarting an index merge operation, leading to an assertion in debug builds. (Bug #27599292)
• Using a binary or hexadecimal literal value in a ENUM column resulted in a string type with bytes not permitted by the data dictionary character set. (Bug #27592803)
• It was possible for CONVERT() to fail under the following conditions:

The size of the result of the conversion was greater than max_allowed_packet

sql_mode was "" (that is, empty).

The character set of the result differed from character_set_connection (Bug #27592714)
• Use of ENUM or SET fields with JSON_TABLE() could result in unexpected behavior. (Bug #27571251)
• Host name resolution errors could cause the audit_log plugin to fail. (Bug #27567003)
• Unsuccessful connection attempts were not being written to the error log when log_error verbosity=3. (Bug #27539838)
• Persisted system variables with a value of NULL were written to mysql-auto.cnf incorrectly. (Bug #27512616)
• An earlier code cleanup caused FEDERATED storage engine failures. (Bug #27493633, Bug #89537)

References: This issue is a regression of: Bug #25943754.
• Setting max_execution_time sometimes had no effect when used with full-text search. (Bug #27155294)
• An attempted read of an uncommitted transaction raised an assertion. (Bug #26876608)
• ALTER TABLE ... REORGANIZE PARTITION ... could result in incorrect behavior if any partition other than the last was missing the VALUES LESS THAN part of the syntax. (Bug #26791931)
• In some cases when ER_DATA_TOO_LONG was the appropriate error, ER_TRUNCATED_WRONG_VALUE_FOR_FIELD was reported instead. (Bug #26475237, Bug #87100)
• A last_checked_for_upgrade column that stores the current server version number was added to the mysql.tables data dictionary table to track successful CHECK TABLE ... FOR UPGRADE operations. This data was previously stored in .frm files to avoid rechecking tables unnecessarily. (Bug #24741307, Bug #83169)
• The MYSQLparse() function generated from the sql_yacc.yy file had exceeded a size limit that caused the Microsoft Visual C++ compiler to abandon optimization. The /d2OptimizeHugeFunctions
compiler switch was enabled to restore optimization (available in Visual Studio 2017 and higher). (Bug #24600301)

- A range check for the product of a signed and unsigned integer could be performed incorrectly. (Bug #17081376, Bug #69715)

- Specifying the maximum possible value for a YEAR column failed when expressed as a real constant such as $2155.0E00$ or $2.15E3$. (Bug #91226, Bug #28172538)

- Prevented potential member access within a null pointer of type SELECT_LEX. (Bug #90689, Bug #27945614)

- The base used for calculation of line numbers reported in warnings generated by SELECT statements has been changed from the number of input rows read to the actual row numbers seen by the user. For statements using GROUP BY, this number is now based on the number of aggregate groupings. (Bug #90667, Bug #27938507)

- Estimates contained by histograms for the number of distinct values in each bucket, which can be employed to estimate the number of distinct values in a table, can now be used by the Optimizer's histograms::Histogram interface. (Bug #90465, Bug #27872281)

- Selecting from the INFORMATION_SCHEMA.KEYWORDS table failed when no default database had been selected. (Bug #90160, Bug #27729859)

- ANALYZE TABLE UPDATE HISTOGRAM failed to return an error condition to the client when lock_wait_timeout was exceeded during statement processing. (Bug #89738, Bug #27574375)

- It was possible to create a table $t$ having a generated column $g$ and then to execute ALTER TABLE $t$ ALTER COLUMN $g$ SET DEFAULT ...; attempting to set the default for a generated column in this fashion now fails with Error 1221 Incorrect usage of DEFAULT and generated column. This is the same behavior as for a similar CREATE TABLE statement. (Bug #88619, Bug #27164393)

### Changes in MySQL 8.0.12 (2018-07-27, General Availability)

- Audit Log Notes
- Compilation Notes
- Configuration Notes
- Data Dictionary Notes
- Function and Operator Notes
- Logging Notes
- MySQL Enterprise Notes
- Optimizer Notes
- Packaging Notes
- Plugin Notes
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
• X Plugin Notes
• Functionality Added or Changed
• Bugs Fixed

Audit Log Notes

• For new MySQL installations, the USER and HOST columns in the audit_log_user table used by MySQL Enterprise Audit now have definitions that better correspond to the definitions of the User and Host columns in the mysql.user system table.

For upgrades to an installation for which MySQL Enterprise Audit is already installed, it is recommended that you alter the table definitions as follows:

```
ALTER TABLE mysql.audit_log_user
   DROP FOREIGN KEY audit_log_user_ibfk_1;
ALTER TABLE mysql.audit_log_filter
   CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_ci;
ALTER TABLE mysql.audit_log_user
   CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_ci;
ALTER TABLE mysql.audit_log_user
   MODIFY COLUMN USER VARCHAR(32);
ALTER TABLE mysql.audit_log_user
   ADD FOREIGN KEY (FILTERNAME) REFERENCES mysql.audit_log_filter(NAME);
```

(Bug #23706056)

Compilation Notes

• **Microsoft Windows:** For OpenSSL 1.1, the library names have changed for Windows. The CMake configuration code now accounts for this. Also, it is now possible on Windows to use `-DWITHOUT_SERVER=1` to build 32-bit client binaries. (Bug #28170711, Bug #91223)

• The minimum version of the Boost library for server builds is now 1.67.0. (Bug #27866110)

Configuration Notes

• The maximum value was increased from 256 to 1024 for these Performance Schema system variables:

```
performance_schema_max_cond_classes
performance_schema_max_file_classes
performance_schema_max_mutex_classes
performance_schema_max_rwlock_classes
performance_schema_max_socket_classes
performance_schema_max_stage_classes
performance_schema_max_thread_classes
```

The default value was increased from 250 to 300 for this Performance Schema system variable:

```
performance_schema_max_mutex_classes
```

(Bug #27647918)

• For RelWithDebInfo builds on Linux, REPRODUCIBLE_BUILD now defaults to ON. (Bug #27483447)

• The default value for the slave_pending_jobs_size_max system variable has increased from 16M to 128M. On a multi-threaded slave (with slave_parallel_workers > 0), this value sets the
maximum amount of memory that is made available to slave worker queues holding events that have not yet been applied. The potentially larger memory usage should be taken into account when calculating the resources required to deploy a multi-threaded slave. Masters and single-threaded slaves are not affected by this setting.

The new default is double the recently increased default value specified by the `max_allowed_packet` system variable (now 64M). The value of `slave_pending_jobs_size_max` is a soft limit, so events larger than its value (consisting of one or more packets) can be queued and processed. However, a large transaction is held until all the slave workers have empty queues, and then processed. All subsequent transactions are held until the large transaction has been completed. The delay to clear the queues of all the slave workers and the wait to queue subsequent transactions can cause lag on the replication slave and decreased concurrency of the slave workers. The value of `slave_pending_jobs_size_max` should therefore be set high enough to accommodate most expected event sizes.

**Data Dictionary Notes**

- The following data dictionary enhancements were introduced:
  - An internal method was added to protect against global read locks. The method is called before storing, updating, or dropping data dictionary objects.
  - Assertion code was added to the data dictionary API to protect against global read locks when storing, updating, or dropping data dictionary objects.
  - Protection is now implicitly acquired against backup locks and global read locks when an exclusive metadata lock is acquired by a storage engine.
  - A cost model cache validity check is now performed before releasing a cost model entry.
  - A data dictionary API function was added for committing and rolling back transactions, and for releasing transactional meta data locks.

(Bug #27937059)

**Function and Operator Notes**

- The `BIT_AND()`, `BIT_OR()`, and `BIT_XOR()` aggregate functions now can be used as window functions, if an `OVER` clause is present. See Aggregate (GROUP BY) Function Descriptions.

**Logging Notes**

- Previously, successful assignments to the `dragnet.log_error_filter_rules` system variable resulted in a Note and a nonzero warning count. To avoid generating a diagnostic for a successful operation, this Note is no longer generated. Instead, the `dragnet.Status` status variable can be consulted to determine the result of the most recent assignment to `dragnet.log_error_filter_rules`. (Bug #27910708, Bug #90571)

- Previously, log components listed in the `log_error_services` system variable had to be separated by semicolons. Components now can be separated by commas. A given setting cannot use both semicolon and comma separators. (Bug #27788925, Bug #90268)

**MySQL Enterprise Notes**

- The MySQL Enterprise Firewall `firewall_whitelist` table now contains a primary key column named `ID`. (Bug #27164826)
Optimizer Notes

- The `filesort` algorithm (used by the optimizer to satisfy an `ORDER BY` clause when no index can be used) is now more memory efficient. The primary change is that the optimizer allocates memory buffers incrementally as needed, up to the size indicated by the `sort_buffer_size` system variable, rather than allocating a fixed amount of `sort_buffer_size` bytes up front. This enables users to set `sort_buffer_size` to larger values to speed up larger sorts, without concern for excessive memory use for small sorts. (This benefit may not occur for multiple concurrent sorts on Windows, which has a weak multithreaded `malloc`.)

  In addition, `filesort` is less pessimistic with respect to worst-case scenarios. Previously, `filesort` produced an error without starting if merge operations or sort-key generation might fail. Now, because the worst case is unlikely to occur in practice, `filesort` produces an error only for actual rather than prospective failures.

- Previously, extended `EXPLAIN` output was produced only for `SELECT` statements. Extended output is now produced for `INSERT`, `REPLACE`, `UPDATE`, and `DELETE` statements as well. (To display extended output, use `SHOW WARNINGS` following `EXPLAIN`. See Extended `EXPLAIN` Output Format.)

Packaging Notes

- For Linux, the generic binary distribution compression algorithm changed from Gzip to XZ; and the file extension changed from `.tar.gz` to `.tar.xz`. (Bug #28450941, Bug #91889)

- Docker disables NUMA-related operations by default, so the `mysqld` included in Docker images now has NUMA support disabled so that the error log does not fill with NUMA warnings. (Bug #28081363)

- RPM `.spec` files were updated to reflect the removal of yaSSL and that OpenSSL is the default SSL library for all builds. (Bug #28025427)

- For generic Linux `.tar.gz` distributions, the `mysqlxtest` client has been moved from the server package to the test package. (Bug #27744137)

- An RPM package for installing ARM 64-bit (`aarch64`) binaries of MySQL Server on Oracle Linux 7 is now available in the MySQL Yum Repository and for direct download.

  Known Limitation for this ARM release: You must enable the Oracle Linux 7 Software Collections Repository (`ol7_software_collections`) to install this package, and must also adjust the `libstdc++7` path. See Yum’s Platform Specific Notes for additional details.

Plugin Notes

- Previously, the `Rewriter` query rewrite plugin permitted rewriting only for `SELECT` statements. Rewritable statements now include `INSERT`, `REPLACE`, `UPDATE`, and `DELETE` as well. See The `Rewriter` Query Rewrite Plugin.

Security Notes

- The linked OpenSSL library for MySQL Server has been updated to version 1.0.2o. Issues fixed in the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html. (Bug #28025379)

Spatial Data Support

- Incompatible Change: The new spatial functions `ST_Longitude()` and `ST_Latitude()` return the longitude or latitude, respectively, of their `Point` argument. If called with a second argument, these functions use the argument to set the longitude or latitude, respectively, of their `Point` argument.
In addition, the \texttt{ST_X()} and \texttt{ST_Y()} functions now interpret their \texttt{Point} argument according to its spatial reference system definition (SRS):

- \texttt{ST_X()} returns the coordinate value of the axis that appears first in the SRS definition, and \texttt{ST_Y()} returns the coordinate value of the axis that appears second in the SRS definition.

- If \texttt{ST_X()} and \texttt{ST_Y()} are used to set the longitude or latitude of a \texttt{Point} argument that has a geographic SRS, the longitude or latitude value must be in the proper range for the SRS, or an error occurs.

\begin{center}
\textbf{Note}
\end{center}

If spatial data contains geometry values that now are interpreted differently by \texttt{ST_X()} and \texttt{ST_Y()}, existing queries using these functions will return different results, compared to previous MySQL versions.

(Bug #27125600, Bug #88503)

- The EPSG Dataset containing spatial reference system data for spatial calculations has been upgraded from version 9.2 to 9.3. (Bug #27977721)

- Previously, the parser recognized \texttt{USING RTREE} in index specifications, but this could not be specified for any storage engine. Now \texttt{USING RTREE} is permitted for spatial index specifications. (Bug #27836608)

- Indexes on geometry columns now must be spatial indexes. Consequently, the \texttt{SPATIAL} keyword is optional but implicit for creating an index on a spatial column. A spatial index on a geometry column is not permitted for a primary key or unique index. Also, \texttt{ASC} and \texttt{DESC} are not permitted for spatial indexes. (Bug #21087676)

\section*{SQL Syntax Notes}

- \textbf{Important Change:} MySQL now allows a query having a \texttt{WITH ROLLUP} modifier to use \texttt{DISTINCT}. See \textit{SELECT Syntax}, for more information. (Bug #87450, Bug #26640100, Bug #20671578, Bug #76229)

- \textbf{Important Change:} MySQL now allows the use of \texttt{ORDER BY} with grouping functions in \texttt{SELECT} statements. (Bug #86312, Bug #27063525)

- \textbf{Important Change:} MySQL now allows \texttt{ORDER BY} to be used in a query having a \texttt{WITH ROLLUP} modifier. For more information, see \textit{SELECT Syntax}. (Bug #86311, Bug #26073513)

- Explicit \texttt{ASC} or \texttt{DESC} qualifiers for \texttt{GROUP BY} clauses are now deprecated and will be removed in a future MySQL version.

\section*{X Plugin Notes}

- Authentication errors returned from a MySQL 8.0.11 server to an X Plugin client were marked as fatal, which caused the client to stop attempting its automatic sequence of authentication attempts using different methods. X Plugin now ignores fatal authentication errors, and subsequent expected errors such as write timeout errors, while the sequence of authentication attempts is in progress. If none of the authentication methods tried are successful, X Plugin now selects and returns the most relevant error message received from the server, which is not necessarily the last error message received. (Bug #28135006)
• During the X Plugin automated authentication sequence, connection attempts are made using different authentication methods in a pre-set sequence. X Plugin checks before starting the sequence whether the connection uses SSL or not, and does not attempt connection with the PLAIN authentication method if the connection is not encrypted. For additional verification, a check and error were added so that the PLAIN authentication method is disallowed if attempted when the connection is not encrypted. (Bug #27691189)

• In the event of an authentication failure, X Plugin always returned the error code ER_ACCESS_DENIED, with a custom X Protocol error message for the error situation. The error codes that are received from the MySQL server for authentication failures are now passed on to the client instead, with the standard MySQL error messages that go with them. (Bug #27675699)

• During the X Plugin automated authentication sequence, where connection attempts are made using different authentication methods in a pre-set sequence, the error returned after every failed connection attempt was marked with the severity FATAL. Now, the errors are marked as ERROR while there are still other authentication methods to be attempted, and only the final error that results in disconnection is marked as FATAL. (Bug #27636947)

• The X Plugin code was being compiled twice with different preprocessor definitions. (Bug #27267054)

• A number of improvements were made to the handling of authentication issues by X Plugin:
  • The X Plugin automated authentication sequence, where connection attempts are made using different authentication methods in a pre-set sequence, is now halted if an error response indicates that the connection has been broken or reads or writes are timing out.
  • The automated authentication sequence is also now halted in the event of any fatal error.
  • The last significant error is now reported for an abandoned authentication sequence, not just the last error that occurred.
  • A more specific error code and message are now reported if authentication with the caching_sha2_password authentication plugin is attempted over an unsecure connection. (Bug #27257774)
  • In some circumstances, a MySQL server with X Plugin loaded could experience an unexpected halt during a shutdown process initiated by a failed start or by the RESTART command. The issue has now been fixed. (Bug #25474793, Bug #27259783)

### Functionality Added or Changed

- **Important Change; Replication:** Use the `group_replication_exit_state_action` option to configure how Group Replication behaves when a server instance leaves the group involuntarily, for example when it is expelled from the group due to an unstable network connection. When `group_replication_exit_state_action` is set to ABORT_SERVER (the default value), the instance shuts itself down, and when `group_replication_exit_state_action` is set to READ_ONLY the instance switches itself to super read only mode instead and goes into the Group Replication ERROR state.

- **InnoDB:** InnoDB now supports ALGORITHM=INSTANT for the following ALTER TABLE operations:
  • Adding a column. This feature is referred to as “Instant ADD COLUMN”.
  • Adding or dropping a virtual column.
  • Adding or dropping a column default value.
• Modifying the definition of an **ENUM** or **SET** column.

• Changing the index type.

• Renaming a table.

Operations that support **ALGORITHM=INSTANT** only modify metadata in the data dictionary. No exclusive metadata locks are taken on the table during preparation and execution phases of the operation, and table data is unaffected, making the operations instantaneous. If not specified explicitly, **ALGORITHM=INSTANT** is used by default by operations that support it. If **ALGORITHM=INSTANT** is specified but not supported, the operation fails immediately with an error.

For more information about operations that support **ALGORITHM=INSTANT**, see Online DDL Operations.

Thanks to the Tencent Games DBA team for the contribution. (Bug #28100103, Bug #91074)

• **InnoDB**: Undo logging is now supported for small updates to large object (LOB) data, which improves performance of LOB updates that are 100 bytes in size or less. Previously, LOB updates were a minimum of one LOB page in size, which is less than optimal for updates that might only modify a few bytes. This enhancement builds upon support added in MySQL 8.0.4 for partial update of LOB data.

• **Replication**: You can now use a shell pipe to supply an archive of compressed binary log files to **mysqlbinlog** as streamed input. Previously, the files had to be extracted separately before passing them to **mysqlbinlog**. Thanks to the Facebook team for this contribution.

In this example, **binlog-files_1.gz** contains multiple binary log files for processing. The pipeline extracts the contents of **binlog-files_1.gz**, pipes the binary log files to **mysqlbinlog** as standard input, and pipes the output of **mysqlbinlog** into the **mysql** client for execution:

```
gzip -cd binlog-files_1.gz | ./mysqlbinlog - | ./mysql -uroot -p
```

(Bug #27445278, Bug #89423)

References: See also: Bug #27836543.

• **Replication**: The **SYSTEM_VARIABLES_ADMIN** and **PERSIST_RO_VARIABLES_ADMIN** privileges have been added to the **mysql.session** reserved user account. (Bug #89873, Bug #27621869)

• **Replication**: Information about XCom's memory usage has been added to the Performance Schema tables. The **setup_instruments** table now has an entry corresponding to the XCom cache, and the **memory_summary_global_by_event_name** table contains the memory usage statistics of the XCom cache.

• **Microsoft Windows**: On Windows, the forking used to implement **RESTART** was suppressible using the **--gdb** option. However, this was a side effect, performed in addition to the other option actions done to set up a debugging environment. In non-debug settings, the new **--no-monitor** option may be used for the sole purpose of suppressing forking the monitor process. For a server started with either **--gdb** or **--no-monitor**, executing **RESTART** causes the server to simply exit without restarting. (Bug #27801043)

• Checking for foreign key relationships by **DROP TABLE** and **DROP DATABASE** was improved. The check now takes place before any tables are deleted, so that in the event of a problem the statement fails before making any changes. Parent and child tables now can be dropped in arbitrary order, as long as they are dropped by the same **DROP TABLE** statement. In addition, error reporting was improved for attempts to drop a parent table without dropping a child table. (Bug #27821060, Bug #17564464, Bug #70531, Bug #22359539, Bug #79610)
• When support for roles was added, the `ADMIN` keyword became a reserved keyword. `ADMIN` is now once again a nonreserved keyword. (Bug #27814204)

• When a client shuts down the server, the server now writes a message to the error log indicating which user performed this action. (Bug #26246628, Bug #86635)

• Previously, for the `--ssl-mode=VERIFY_IDENTITY` option, the client checked whether the host name that it used for connecting matched the Common Name value in the certificate but not the Subject Alternative Name value. Now, if the client uses OpenSSL 1.0.2 or higher, the client checks whether the host name matches either the Subject Alternative Name value or the Common Name value in the server certificate. Thanks to Daniël van Eeden for a patch on which this change was based. (Bug #16211011, Bug #68052, Bug #27511233, Bug #89578)

Bugs Fixed

• **Important Change; JSON:** The `JSON_TABLE()` function now automatically decodes base-64 values and prints them using the character set given by the column specification. (Bug #90157, Bug #27729112)

  References: See also: Bug #89847, Bug #27613276.

• **InnoDB:** New columns added to `INFORMATION_SCHEMA.INNODB_TABLES` and `INFORMATION_SCHEMA.INNODB_COLUMNS` tables in 8.0.12 did not appear after upgrading to 8.0.12. (Bug #28065244)

• **InnoDB:** Large object (LOB) version numbers stored in LOB index entries were left uninitialized at 0 instead of initialized at 1. (Bug #28046298)

• **InnoDB:** Adding a column to a table with a generated column raised an invalid assertion. (Bug #28040201)

• **InnoDB:** An attempt to access freed memory during a checkpoint operation caused the server to exit. (Bug #28039477)

• **InnoDB:** An assertion failure occurred when starting the server with experimental system variables that were unintentionally exposed in normal MySQL builds. (Bug #27899424, Bug #90526, Bug #27898396, Bug #90523, Bug #27898344, Bug #90522)

• **InnoDB:** An invalid system variable setting raised an assertion. (Bug #27898284)

• **InnoDB:** Attempting to move a table from a discarded file-per-table tablespace to a shared tablespace raised an assertion. The operation now produces an error instead. (Bug #27861972)

• **InnoDB:** Potential failures related to page allocation for large objects (LOBs) were not sufficiently handled. (Bug #27852003)

• **InnoDB:** The ngram full-text search parser permitted comma and period characters to be tokenized as words, which caused an inconsistency between boolean and natural language mode search results. Comma and period characters are no longer tokenized. (Bug #27847697)

• **InnoDB:** Checksum mismatches and other issues were caused by concurrent modification of a tablespace header page, and attempting to place a lock on a partially initialized page. (Bug #27823064)

• **InnoDB:** An I/O error returned by an `fsync()` operation is now treated as a hard error. (Bug #27805553, Bug #90296)

• **InnoDB:** In a single thread, two minitransactions latched the first page of a large object value (LOB) while another thread attempted to allocate the page, causing a deadlock. (Bug #27777959)
• **InnoDB**: After upgrading to MySQL 8.0.12, table columns including the FTS_DOC_ID column were missing from the `INFORMATION_SCHEMA.INNODB_COLUMNS` table. (Bug #27774145)

• **InnoDB**: An invalid assertion was raised when the optimizer chose a prefix index defined on a virtual column. (Bug #27755892)

  References: This issue is a regression of: Bug #19806106.

• **InnoDB**: An in-memory table object for a table being truncated could be opened under improper metadata locking protection.

  A new method of preserving the `AUTOINC` counter value during a `TRUNCATE PARTITION` operation was implemented. (Bug #27754995)

• **InnoDB**: An assertion was raised during a mini-transaction commit that included dirty pages but no redo log records. The mini-transaction did not wait for space in the recently-closed buffer and subsequently marked the page with an invalid current LSN value. After the invalid LSN value was added to the flush list, a delayed mini-transaction added dirty pages with an earlier LSN value, breaking the flush list order.

  The `innodb_log_checkpoint_fuzzy_now` debug option was added. Enabling this option forces InnoDB to write a fuzzy checkpoint. Also, new internal redo log module counters were added for tracking LSN values. The `log_lsn_buf_dirty_pages_added` counter tracks the LSN value up to which dirty pages have been added, and the `log_lsn_buf_pool_oldest_lwm` counter tracks the low watermark LSN for the oldest modified block in the buffer pool. (Bug #27664539)

• **InnoDB**: The Variance-Aware Transaction Scheduling algorithm (VATS) tracks the number of transactions waiting for another transaction. In the case of a deadlock, the method by which the number of waiting transactions is approximated produced inflated values. (Bug #27646322)

• **InnoDB**: Attempting to acquire an InnoDB ticket while holding a victim transaction lock caused a lock order violation. (Bug #27626681)

  References: This issue is a regression of: Bug #23476050.

• **InnoDB**: An attempt to read a zero-length BLOB value raised an assertion failure during recovery. The BLOB value was not yet completely inserted. (Bug #27617389)

• **InnoDB**: A race condition occurred when acquiring a table lock for a table that was already in memory. (Bug #27586419, Bug #27577704)

  References: This issue is a regression of: Bug #26848711.

• **InnoDB**: After restarting the server, attempting to create an encrypted table returned a generic error from the storage engine. (Bug #27577339)

• **InnoDB**: The use of Variance-Aware Transaction Scheduling (VATS) with a spatial index caused a transaction to wait indefinitely. (Bug #27572937)

• **InnoDB**: Incorrect parsing of a tablespace name that included multibyte characters caused a `DROP DATABASE` operation to fail. (Bug #27566937)

• **InnoDB**: A check performed to identify temporary tables caused an error when the data dictionary attempted to open a partitioned table. (Bug #27565997)

• **InnoDB**: A schema mismatch error reported during an import tablespace operation failed to print mismatched table flags in a readable format. (Bug #27542720)

• **InnoDB**: A DDL operation failed to wait for a `FULLTEXT` index optimization operation to finish. (Bug #27326796)
References: This issue is a regression of: Bug #24938374.

- **InnoDB**: An unnecessary check for read-only transactions was removed from the `trx_set_rw_mode()` function. Thanks to Sandeep Sethia for the patch. (Bug #27211287, Bug #88739)
- **InnoDB**: A DDL operation that added a foreign key constraint raised an assertion when it accessed a stale memory object that belonged to the parent table. (Bug #27208858)
- **InnoDB**: A failing assertion occurred after initiating a `memcached` get operation. (Bug #26876594)
- **InnoDB**: A corrupt index ID encountered during a foreign key check raised an assertion. (Bug #26654685)
- **InnoDB**: A `DROP TABLE` operation on a table residing in a general tablespace raised an assertion when checking index status in the post-DDL phase of the operation. (Bug #26523254)
- **InnoDB**: An internal deadlock during a DDL operation resulted in a long semaphore wait followed by a server exit. (Bug #26225783)
- **InnoDB**: A DDL operation encountered a serious error due to an invalid lock upgrade. (Bug #26225783)
- **InnoDB**: On a Windows 64-bit system, invalid buffer pool configuration values caused the server to exit on startup. (Bug #26100239, Bug #86370)
- **InnoDB**: A debug code block related to mutex type definitions was removed. It no longer differed from the non-debug version of the code block. (Bug #24952279, Bug #83529)
- **InnoDB**: A server failure during recovery occurred due to a log parsing buffer overflow. Space is now allocated to the log parsing buffer dynamically and incremented as necessary until it reaches the size specified by `innodb_log_buffer_size`. (Bug #24734190)
- **InnoDB**: Table locking was optimized for concurrent updates on the same table by avoiding costly iterations over the table locks queue. Thanks to Zhai Weixiang for the contribution. (Bug #18955152, Bug #72948)

- **Packaging**: On Windows, the `mysqld.exe --help` command displayed unnecessary and unexpected details in the system output, which this fix eliminates. (Bug #27894020)
- **Partitioning**: Rollback of `ALTER TABLE ... TRUNCATE PARTITION` statements was not handled correctly. (Bug #27603025)

References: See also: Bug #87562, Bug #26710839.

- **Partitioning**: For a partitioned table, partition update time could be incorrect after rebuilding the table or restarting the server. (Bug #27073100)
- **Replication**: If the option `--skip-log-bin` was specified at server startup to disable binary logging, no warning message was issued if both `expire_logs_days` and `binlog_expire_logs_seconds` were set to a nonzero value. The `--skip-log-bin` option meant there were no binary log files to expire, so the outcome of these binary log expiration period settings (which is that the `expire_logs_days` value is ignored) had no practical effect. However, the warning message is now issued, so that the situation can be corrected before binary logging is enabled for the server. (Bug #27699608)
- **Replication**: The log messages generated when a member fails to join the group have been improved, for example when `group_replication_group_name` on the member joining the group does not
match the seed's `group_replication_group_name` this is now described in the log message. (Bug #27628695)

- **Replication**: If a MySQL server is started in crash recovery mode with an `innodb_force_recovery` setting of 4 or greater, which places InnoDB into read-only mode, the semisynchronous replication master plugin should not be initialized. Due to a change in plugin initialization order in MySQL 8.0, the server assumed that if the plugin was registered, it had also been initialized, and incorrectly attempted to deinitialize it. The handling of this plugin in crash recovery mode has now been corrected so that the deinitialization function is not executed when the plugin has not been initialized. (Bug #27481872)

- **Replication**: The use of replication filters or binary log filters can cause issues when they are applied to tables that are updated with XA transactions. Filtering of tables could cause an XA transaction to be empty on a replication slave, and empty XA transactions are not supported. Also, with the settings `master_info_repository=TABLE` and `relay_log_info_repository=TABLE` on a replication slave, which became the defaults in MySQL 8.0, the internal state of the data engine transaction is changed following a filtered XA transaction, and can become inconsistent with the replication transaction context state.

  Due to these issues, the use of replication filters or binary log filters in combination with XA transactions is not supported. This fixes the new error `ER_XA_REPLICATION_FILTERS`, which is logged whenever an XA transaction is impacted by a replication filter, whether or not the transaction was empty as a result. If the transaction is not empty, the replication slave is able to continue running, but you should take steps to discontinue the use of replication filters with XA transactions in order to avoid potential issues. If the transaction is empty, the replication slave stops. In that event, the replication slave might be in an undetermined state in which the consistency of the replication process might be compromised. In particular, the `gtid_executed` set on a slave of the slave might be inconsistent with that on the master. To resolve this situation, isolate the master and stop all replication, then check GTID consistency across the replication topology. Undo the XA transaction that generated the error message, then restart replication. (Bug #27442477)

- **Replication**: When `mysqlbinlog` was used to read a binary log event larger than 128KB, the read failed because 128KB was returned as the size, instead of the actual size. The issue has now been fixed. Thanks to the Facebook team for the patch. (Bug #27417084, Bug #89326)

- **Replication**: When a transaction larger than the binary log transaction cache size (`binlog_cache_size`) was flushed to a temporary file during processing, and the flush failed due to a lack of space in the temporary directory, the flush error was not handled correctly. No message was written to the error log, and the binary log cache was not cleared after the transaction was rolled back. Now, in this situation, the server takes an appropriate action based on the `binlog_error_action` setting (shut down the server or halt logging), and writes a message to the error log. When the transaction is rolled back, the server checks for flush errors and clears the binary log cache if any occurred. (Bug #27399620, Bug #89272)

- **Replication**: When GTIDs are in use for replication, replicated transactions that are filtered out on the slave are persisted. If binary logging is enabled on the slave, the filtered-out transaction is written to the binary log as a `Gtid_log_event` followed by an empty transaction containing only `BEGIN` and `COMMIT` statements. If binary logging is disabled, the GTID of the filtered-out transaction is written to the `mysql.gtid_executed` table. This process ensures that there are no gaps in the set of executed GTIDs, and that the filtered-out transactions are not retrieved again if the slave reconnects to the master. Previously, this process was not done for `CREATE DATABASE`, `ALTER DATABASE`, and `DROP DATABASE` statements, but it is now carried out for those statements as well as for others. (Bug #27308751, Bug #88891)

- **Replication**: On a multithreaded slave, when a `STOP SLAVE` statement is executed on the slave, followed by a `START SLAVE` statement, the error log can report a different position in the binary log for
the slave SQL thread when exiting, compared to the position reported for the slave SQL thread at the subsequent initialization.

For a multithreaded slave, the position reported for the SQL thread on exit is a low water mark, up to which the replication stream is consistent and has no gaps. Transactions appearing before the position are guaranteed to have committed, but transactions after the position may have committed or not. However, this low water mark was being reported before the process to stop the worker threads was actually carried out, and the low water mark was subsequently updated by a checkpoint routine during that process. The timing of the log message has now been changed so that the final low water mark is reported as the position for the SQL thread on exit. (Bug #27300658)

- **Replication:** An online upgrade of a replication group that was running a version of MySQL earlier than 5.7.23 was not possible. When a member running a version earlier than either MySQL 5.7.22 or MySQL 8.0.11 tried to join a group with members running MySQL 5.7.21 or earlier, it failed to join the group because MySQL 5.7.21 does not send the value of the `lower_case_table_names` system variable. This means that the later version compared the `lower_case_table_names` variable with an invalid value and exited the group. The workaround is to take the group offline, upgrade to version 5.7.23 or 8.0.12 and later. Subsequent upgrades of the group can then be carried out without taking the group offline. See [Upgrading Group Replication](#). (Bug #90794, Bug #27991334)

- **Replication:** The `gtid_purged` variable contains the binary log events that have been purged from the log by the server. When running the Group Replication plugin this variable should never be modified, but it was possible. Now the variable cannot be modified when Group Replication is running. (Bug #90146, Bug #27724561)

- **Replication:** The `PIT_TRANSACTIONS_NEGATIVE_CERTIFIED`, the `PIT_TRANSACTIONS_ROWS_VALIDATING` and the `PIT_TRANSACTIONS_LOCAL_ROLLBACK` member messages were not being correctly decoded. (Bug #90077, Bug #27692831)

- **Replication:** In certain situations, such as during distributed recovery procedure, the certification info garbage collection was purging more data than it should, resulting in conflicts not being detected. The garbage collection procedure has been improved to take this case in consideration. (Bug #89938, Bug #27652526)

- **Replication:** The `ER_GRP_RPL_SQL_SERVICE_FAILED_TO_RUN_SQL_QUERY` error was being logged incorrectly. (Bug #89788, Bug #27590534)

- **Replication:** When the `group_replication_applier` channel's applier thread encountered an error, the `master_log_name` and `end_log_pos` in the error message were incorrect. In Group Replication, the events of a transaction are replicated before they are written to the binary log of the member where the transaction originated. The result is that the final `master_log_name` and the `end_log_pos` of those events are unknown at the time they are applied on the replica by `group_replication_applier` channel's applier thread. To avoid confusion, now any such error messages encountered by a `group_replication_applier` channel do not contain the binary log name and the binary log position. (Bug #89146, Bug #27368735)

- **Replication:** Using `group_replication_force_members` to unblock a group, for example after losing majority, sometimes failed with error 1231. (Bug #86957, Bug #26394418)

- **macOS:** On macOS, the server autostarted at system boot time even when set not to in the MySQL preference pane. (Bug #27969174, Bug #90672)

- **Microsoft Windows:** For debug builds on Windows, `CREATE SPATIAL REFERENCE SYSTEM` with an empty definition string could raise assertions. (Bug #27672683)

- In the client library, signals were incorrectly unblocked for `ppoll()`. Thanks to Facebook for the patch. (Bug #28075623, Bug #90999, Bug #28096808, Bug #91067)
• Certain `ALTER TABLE` statements could disable table encryption. (Bug #28045585)

  References: This issue is a regression of: Bug #27389878.

• Under some conditions, `MyISAM` code compilation failed due to a missing include file. (Bug #28039150, Bug #90898)

• The `REGEXP_REPLACE()` function truncated its result when used in an `UPDATE` statement. (Bug #28027093, Bug #90870)

  `REGEXP_REPLACE()` results from one result set row could carry forward to the next row, resulting in accumulation of previous results in the current row. (Bug #27992118, Bug #90803)

• A heap overflow vulnerability in the MySQL client library was fixed. (Bug #27980823)

• Users who had the `SUPER` privilege were not permitted to modify the `keyring_operations` system variable. (Bug #27976270)

• Full-text search auxiliary tables created in MySQL 5.7 and MySQL 8.0.11 with timestamps of 0 caused a data dictionary upgrade failure when upgrading from MySQL 8.0.11 to MySQL 8.0.12. (Bug #27960500)

• A “source and destination overlap in memcpy” Valgrind error was returned when upgrading from MySQL 5.7. (Bug #27945658, Bug #90691)

• Linux binaries were unable to dump core after executing a `setuid()` call. (Bug #27929894, Bug #90642, Bug #23337428, Bug #21723)

  `RESET PERSIST` did not flush variable values from memory. (Bug #27924206)

• For debug builds, using `CREATE TABLE ... SELECT` to create a table with a long numeric string and indexing the column could lead to an assertion being raised. (Bug #27909771)

• For debug builds, `RESET PERSIST` for an unknown variable could cause an assertion to be raised. (Bug #27903874, Bug #90546)

  References: This issue is a regression of: Bug #27374791.

• Updates to column metadata of a view referencing a partitioned table could cause a server exit. (Bug #27903842)

• For debug builds, an assertion could be raised for `UNION` queries when computing the combined data type of a `GEOMETRY` column and `SELECT * FROM (SELECT NULL)`. (Bug #27903792, Bug #90550)

• For debug builds, starting the server with `thread_stack=0` and executing `ALTER TABLE` could raise an assertion. (Bug #27899274, Bug #90525)

  `SHOW TABLE STATUS` could cause a server exit. (Bug #27897815)

• For generated columns that used the `INTERVAL()` function, incorrect behavior could occur. (Bug #27881102)

• For Fedora 28, MySQL Community RPM packages would not install if Connector ODBC native packages were installed. (Bug #27850721)

• It was possible to drop the Performance Schema. (Bug #27830283)

• A runtime error could occur for calls to `mysql_sys_var_longlong()` or `String::copy()`. (Bug #27822413, Bug #27808412)
• An assertion was raised by error logging code that used an inappropriate error number. (Bug #27792903)

  References: This issue is a regression of: Bug #27462408.

• The `CONVERT TO CHARACTER SET charset_name` clause for `ALTER TABLE` produced a syntax error if `charset_name` was `DEFAULT`. (Bug #27760787, Bug #90222)

  References: This issue is a regression of: Bug #27389878.

• An unencrypted connection could result from a client connection attempt specifying that an encrypted connection was required, if the server was not configured to support SSL. (Bug #27759871)

• `INSERT INTO ... SELECT` did not perform as many checks as `INSERT INTO ... VALUES()` with respect to requiring valid geometry values for geometry columns. (Bug #27756083)

• `REGEXP_INSTR()` did not return an error message when used with invalid match mode flags. (Bug #27751277)

• Some data dictionary table definitions were modified to align more closely with their initial design. Additionally, the data dictionary version number was incremented, and code that refers to data dictionary table indexes now uses symbolic index names instead of numeric values. (Bug #27745526, Bug #90196)

• Executing a prepared statement to insert rows with large number of parameters was very slow. (Bug #27699248)

• `ibd2sdi` could exit due to performing a divide-by-zero operation. (Bug #27692051, Bug #90071)

• The `mysql-boost-8.0.4.tar.gz` source distribution omitted `NDB` sources. (Bug #27690232)

• `SHOW GRANTS` could cause a server exit if executed for an unprivileged user who inherited privileges from an anonymous user. (Bug #27678129)

• The parser incorrectly permitted the index name for `CREATE INDEX` to be empty. (Bug #27676427)

• Failures of server restarts initiated by systemd or `mysqld_safe` were not written to the error log. (Bug #27675050, Bug #90012)

• For debug builds, if a transaction was in read-only mode, trying to create or drop a histogram could raise an assertion. (Bug #27672693)

• If the server was started with `--skip-grant-tables, SET ROLE DEFAULT` caused a server exit. (Bug #27672154)

• Casting of `VARBINARY` values to `TIME` values was not performed correctly. (Bug #27665997)

• Error checking was not performed after creation of a condition for a `const` reference. (Bug #27665085)

• The dictionary information object managed by the MeCab model was used after the MeCab model was released. (Bug #27660368)

• Attempting to abort events of the `MYSQL_AUDIT_AUTHENTICATION_CLASS` class caused a server exit. These events are now not abortable. (Bug #27645636)

• These resource-group statements are now disallowed in stored functions because they cause an implicit commit, which is not permitted in a stored function: `CREATE RESOURCE GROUP, ALTER RESOURCE GROUP, DROP RESOURCE GROUP`. (Bug #27638623, Bug #89914)

• Audit log filter rules did not permit class names to be specified as an array of strings. That is now possible. Example:
(Bug #27628325)

- On Windows, if the Visual C++ Redistributable for Visual Studio had been removed, MySQL uninstallation using the MSI installer failed. (Bug #27621546)

- `mysql_secure_installation` now loads the `validate_password` component rather than the deprecated `validate_password` plugin. (Bug #27619667)

- `REGEXP_LIKE()` using an expression containing a question mark (?) was not handled correctly when included in a prepared statement. (Bug #27595368)

- `REGEXP_SUBSTR()` did not always handle NULL patterns correctly. (Bug #27572258)

- Some `SHOW CREATE TABLE` statements could raise an assertion. (Bug #27569314)

- Spurious stages could be seen in the Performance Schema `events_stages_current` table. (Bug #27566220)

- Triggers for a table could execute in an order different from that specified by their `PRECEDES/FOLLOWS` clauses. (Bug #27544152)

- Attempts to set some persisted variables at startup (such as `keyring_operations` and `mandatory_roles`) produced a privilege violation error and the server did not start or did not set the variable. (Bug #27523095)

- Under some conditions, an unused-variable warning occurred in the MeCab full-text search code. Thanks to Laurynas Biveinis for the patch. (Bug #27519952, Bug #89598)

- Within stored procedures, it was not possible to write a condition handler to catch foreign key failures if the table involved used the `FEDERATED` storage engine. (Bug #27509959)

- It was possible to drop a primary key that served as the parent key in a foreign key relationship. In addition, for table creation, the SQL layer now checks whether there is a parent key for the foreign key and reports an error if it is missing. (Bug #27506922, Bug #89570, Bug #11754696, Bug #46337)

- With the `NO_ENGINE_SUBSTITUTION` SQL mode disabled, an error (rather than substitution) occurred for `CREATE TABLE` and `ALTER TABLE` if the desired engine was disabled using the `disabled_storage_engines` system variable. (Bug #27502530)

- Key rotation using the `keyring_encrypted_file` keyring plugin could cause a server exit. (Bug #27497018)

- If MySQL Enterprise Firewall was configured with a large number of rules, server shutdown could take a long time. (Bug #27492122)

- `SET PERSIST_ONLY var_name = DEFAULT` persisted the current global value, not the default value. (Bug #27489026)

- If MySQL is configured with the `LINK_RANDOMIZE` CMake option enabled, the `--gc-sections` compiler option is now disabled; the combination causes segmentation faults. (Bug #27484106)
• During server installation, the `mysql.infoschema`, `mysql.session`, and `mysql.sys` reserved users were associated with the `mysql_native_password` authentication plugin even when the default plugin was `caching_sha2_password`. (Bug #27454299)

• Some window functions used the wrong precision for their aggregated data types. (Bug #27452179)

• For a hexadecimal value $H$, `SELECT with WHERE (-1) - H` raised an assertion. (Bug #27452082)

  References: See also: Bug #27041382. This issue is a regression of: Bug #21982792.

• A memory leak in the `pfs-t` unit test was fixed. Thanks to Yura Sorokin for the patch. (Bug #27440735, Bug #89384)

• Multiple invocations of `CHAR()` within a stored program could yield different results once an invalid code point was passed as the argument. (Bug #27410088)

• With `automatic_sp_privileges` enabled, the `EXECUTE` and `ALTER ROUTINE` privileges were not correctly granted to routine creators. (Bug #27407480)

• Some queries that used ordering could access an uninitialized column during optimization and cause a server exit. (Bug #27389294)

• For debug builds, an assertion was raised (rather than an error) for `ALTER TABLE` for an `ARCHIVE` table with a `GEOMETRY NOT NULL` column containing empty strings. (Bug #27330634, Bug #89088)

• Accounts that authenticated with the `auth_sock` authentication plugin could not connect using older clients. (Bug #27306178)

• If system tables could not be read or initialized, the server wrote an incomplete error message. Thanks to Daniël van Eeden for a patch related to the fix. (Bug #27302337, Bug #89001)

• `RENAME USER` failed when used to rename a role that was not granted to any account or role. (Bug #27284699)

• The `keyring_okv` plugin did not always fail over to the standby OKV server when the primary server was unavailable. (Bug #27244099)

• The Performance Schema could attempt to access invalid `SOURCE` column values. (Bug #27231036)

• Mishandling of internal privilege structures could cause a server exit. (Bug #27230925)

• For debug builds, use of a too-small type for calculating the determinant in GIS computations could cause an assertion to be raised. (Bug #27135504)

• For attempts to increase the length of a `VARCHAR` column of an `InnoDB` table using `ALTER TABLE` with the `INPLACE` algorithm, the attempt failed if the column was indexed. If an index size exceeded the `InnoDB` limit of 767 bytes for `COMPACT` or `REDUNDANT` row format, `CREATE TABLE` and `ALTER TABLE` did not report an error (in strict SQL mode) or a warning (in nonstrict mode). (Bug #26848813)

• In MySQL 5.7.19, the `mysql.gtid_executed` table was excluded from dumps. This table is no longer excluded but its data is not dumped. (Bug #26643180, Bug #87455)

  References: This issue is a regression of: Bug #82848, Bug #24590891.

• The `slave_rows_search_algorithms` system variable controls how rows are searched for matches when preparing batches of rows for row-based logging and replication. Specifying `INDEX_SCAN` as one of the search algorithms performs an index scan if an index is present. In the situation where a different primary key is used on the master and the slave, and a unique key is present on the slave, a bug in
the code meant that the index scan was not being performed as it should be, and a slower table scan was being performed instead. The issue has now been corrected so that an index scan is used. (Bug #26450129, Bug #23311892, Bug #81500, Bug #81501)

- Certain cases of subquery materialization could cause a server exit. These queries now produce an error suggesting that materialization be disabled. (Bug #26402045)

- Spatial operations on geographic GeometryCollection values could return ER_GIS_INVALID_DATA for valid geometries. (Bug #26174808)

- For MyISAM tables, particular sequences of INSERT and DELETE statements could cause table corruption. (Bug #25541037)

- The SHOW CREATE USER statement requires the SELECT privilege for the mysql database, except to see information for the current user. In addition, for the current user, the statement requires the SELECT privilege for the mysql.user system table for display of the password hash in the IDENTIFIED AS clause; otherwise, the hash displays as <secret>. (Bug #24911117)

- CREATE FUNCTION produced an error if there was no default database and the function body called another function. (Bug #24357244, Bug #82350)

- For debug builds, a SELECT statement executed within a stored program could raise an assertion if derived table resolution failed. (Bug #23221336)

- If mysqldump or mysqlpump were used to dump binary data without the --hex-blob option, reloading the dump file could produce spurious warnings (values were inserted correctly regardless of the warnings). Such values are now written preceded by the _binary introducer to silence the warnings. (Bug #22601255, Bug #80150)

- String comparisons involving wildcards could reference uninitialized memory. (Bug #12635103)

- REGEXP_REPLACE() raised a Valgrind warning when invoked with an invalid capture group. (Bug #87842, Bug #27612255)

Changes in MySQL 8.0.11 (2018-04-19, General Availability)

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downgrade from MySQL 8.0 to MySQL 5.7 (or from a MySQL 8.0 release to a previous MySQL 8.0 release) is not supported. The only supported alternative is to restore a backup taken before upgrading.</td>
</tr>
</tbody>
</table>

- Character Set Support
- Compilation Notes
- Configuration Notes
- Deprecation and Removal Notes
- Error Handling
- INFORMATION_SCHEMA Notes
- Installation Notes
- Logging Notes
- Packaging Notes
Character Set Support

- The utf8mb3 character set will be replaced by utf8mb4 in a future MySQL version. The utf8 character set is currently an alias for utf8mb3, but will at that point become a reference to utf8mb4. To avoid ambiguity about the meaning of utf8, consider specifying utf8mb4 explicitly for character set references instead of utf8.

Compilation Notes

- MySQL source code has been converted to Google style rules using clang-format. (Bug #27414321)
- Sun RPC is being removed from glibc. CMake now detects and uses libtirpc if glibc does not contain Sun RPC. (You might find it necessary to install libtirpc and rpcgen to take advantage of this CMake feature.) (Bug #27368272, Bug #89168)
- The minimum version of the Boost library for server builds is now 1.66.0. (Bug #27137349, Bug #88540)
- When building from source, the provided CMake script assumed that JUnit and Ant were installed and available at set paths. If this was not the case, building the GCS Java wrapper resulted in build errors. To address this a new CMake option WITH_ANT=path_name has been added to the build script, and which works in a similar way to the existing WITH_BOOST CMake option. Set WITH_ANT to the path of a directory where the Ant tarball, or an already unpacked archive, is saved. When WITH_ANT is not set, or is set with the special value system, the build assumes a binary ant exists in $PATH.

Similarly, the CMake option ENABLE_DOWNLOADS already exists to download required libraries during a build, such as the Google Test distribution. Now, when ENABLE_DOWNLOADS is set, and JUnit or Ant are not found they are downloaded as part of the build process. (Bug #26678793)

Configuration Notes

- A new CMake option, BUNDLE_RUNTIME_LIBRARIES, enables bundling of runtime libraries with server MSI and Zip packages for Windows. (Bug #27584796, Bug #89771)
- systemd service files now increase LimitNOFILE to 10000, to accommodate higher default settings in MySQL 8.0. (Bug #27530282)
- The new WITH_SYSTEM_LIBS CMake option, when enabled, selects the system value for any of the following options that are not set explicitly: WITH_EDITLINE, WITH_ICU, WITH_LIBEVENT, WITH_LZ4, WITH_PROTOBUF, WITH_SSL, WITH_ZLIB. (Bug #27356668, Bug #89136)
- The version of the compiled-in zlib library is now available at runtime as the value of the version_compile_zlib system variable. Thanks to Daniël van Eeden for the patch. (Bug #27235065, Bug #88825)
Deprecation and Removal Notes

- **Incompatible Change**: These deprecated compatibility SQL modes have been removed: **DB2, MAXDB, MSSQL, MYSQL323, MYSQL40, ORACLE, POSTGRESQL, NO_FIELD_OPTIONS, NO_KEY_OPTIONS, NO_TABLE_OPTIONS**. They can no longer be assigned to the `sql_mode` system variable or used as permitted values for the `mysqldump --compatible` option.

  Removal of **MAXDB** means that the `TIMESTAMP` data type for `CREATE TABLE` or `ALTER TABLE` is treated as `TIMESTAMP`, and is no longer treated as `DATETIME`.

  For MySQL 5.7 applications that use SQL modes removed in MySQL 8.0, statements may fail when replicated from a MySQL 5.7 master to a MySQL 8.0 slave, or may have different effects on master and slave. To avoid such problems, applications that use modes removed in MySQL 8.0 should be revised to avoid them.

- The following features related to account management have been removed:
  - Using `GRANT` to create users. Instead, use `CREATE USER`. Following this practice makes the `NO_AUTO_CREATE_USER` SQL mode immaterial for `GRANT` statements, so it too is removed.
  - Using `GRANT` to modify account properties other than privilege assignments. This includes authentication, SSL, and resource-limit properties. Instead, establish such properties at account-creation time with `CREATE USER` or modify them afterward with `ALTER USER`.
  - `IDENTIFIED BY PASSWORD 'hash_string'` syntax for `CREATE USER` and `GRANT`. Instead, use `IDENTIFIED WITH auth_plugin AS 'hash_string'` for `CREATE USER` and `ALTER USER`, where the `'hash_string'` value is in a format compatible with the named plugin.

    Additionally, because `IDENTIFIED BY PASSWORD` syntax has been removed, the `log_builtin_as_identified_by_password` system variable is superfluous and has been removed.

  - The `PASSWORD()` function. Additionally, `PASSWORD()` removal means that `SET PASSWORD ... = PASSWORD('auth_string')` syntax is no longer available.

  - The `old_passwords` system variable.

    For MySQL 5.7 applications that use features removed in MySQL 8.0, statements may fail when replicated from a MySQL 5.7 master to a MySQL 8.0 slave, or may have different effects on master and slave. To avoid such problems, applications that use features removed in MySQL 8.0 should be revised to avoid them and use alternatives when possible. (Bug #27455708)

Error Handling

- `perror` is more flexible in understanding its arguments. For example, for the `ER_WRONG_VALUE_FOR_VAR` error, `perror` understands any of these arguments: `1231`, `001231`, `MY-1231`, or `MY-001231`, or `ER_WRONG_VALUE_FOR_VAR`. If an error number is in the range where MySQL and operating system errors overlap, `perror` displays both error messages. (Bug #27462408)

INFORMATION_SCHEMA Notes

- The new `INFORMATION_SCHEMA.KEYWORDS` table lists the words considered keywords by MySQL and, for each one, indicates whether it is reserved. This table provides applications a runtime source of MySQL keyword information. See The INFORMATION_SCHEMA KEYWORDS Table.
In addition, if you build MySQL from source, the build process generates a `keyword_list.h` header file containing an array of keywords and their reserved status. This file can be found in the `sql` directory under the build directory. This file may be useful for applications that require a static source for the keyword list.

### Installation Notes

- RPM and Debian packages now add the current MySQL series to the `share` directory name (for example `share/mysql-8.0`). (Bug #27638192, Bug #26990531)

### Logging Notes

- Messages written to the error log now indicate the subsystem in which the event occurred. Possible subsystem values are InnoDB (the InnoDB storage engine), Repl (the replication subsystem), Server (otherwise).

### Packaging Notes

- Installing or upgrading using Debian packages now presents a dialog with an option for setting the default authentication plugin to `mysql_native_password`. This may be useful for backward compatibility at some installation sites. (Bug #27454287, Bug #27667294)
- The `compile_innodb` script is no longer used and has been removed from MySQL distributions. Thanks to Alexey Kopytov for the suggestion. (Bug #27448099, Bug #89437)

### Performance Schema Notes

- A new Performance Schema table `log_status` provides information that enables an online backup tool to copy the required log files without locking those resources for the duration of the copy process. When the `log_status` table is queried, the server blocks logging and related administrative changes for just long enough to populate the table, then releases the resources. The `log_status` table informs the online backup which point it should copy up to in the master's binary log and `gtid_executed` record, and the relay log for each replication channel. It also provides relevant information for individual storage engines, such as the last log sequence number (LSN) and the LSN of the last checkpoint taken for the InnoDB storage engine.

  The `BACKUP_ADMIN` privilege is required for access to the `log_status` table. (Bug #27599831)

- Performance Schema objects now have a default character set of `utf8mb4`. (Bug #27407745)

### Security Notes

- yaSSL is no longer included in MySQL source distributions. wolfSSL may be used as a functionally equivalent alternative that has a GPLv2-compatible license. In addition, wolfSSL (like OpenSSL) supports the TLSv1.2 protocol, which yaSSL does not.

  For instructions about building MySQL from source using wolfSSL, see the `extra/README-wolfssl.txt` file in a MySQL source distribution. The `WITH_SSL` CMake option now permits `wolfssl` as a valid value, and no longer takes `bundled` as a value. To use `-DWITH_SSL=wolfssl`, you must follow the instructions in `README-wolfssl.txt`. For more information, see SSL Library-Dependent Capabilities. (Bug #26671671)

- MySQL now supports FIPS mode, if compiled using OpenSSL, and an OpenSSL library and FIPS Object Module are available at runtime. FIPS mode imposes conditions on cryptographic operations such as restrictions on acceptable encryption algorithms or requirements for longer key lengths. The
ssl_fips_mode system variable enables control of FIPS mode on the server side. The `--ssl-fips-mode` client option enables control of FIPS mode on the client side for these programs: `mysql`, `mysqladmin`, `mysqlbinlog`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlpump`, `mysqlshow`, `mysqslap`, `mysqltest`, `mysqlxtest`, `mysql_secure_installation`, and `mysql_upgrade`. See FIPS Support.

Spatial Data Support

- **Incompatible Change:** These geometry collection changes have been implemented:
  - The standard `GeomCollection` name for the geometry collection data type is supported as a synonym for the `GeometryCollection` data type name.
  - The `GeomCollection()` function is supported as a synonym for the `GeometryCollection()` function.
  - `GeomCollection` and `GeomCollection()` are the preferred data type and function names.

Incompatibility: As a consequence of the change in preferred data type name, some `INFORMATION_SCHEMA` tables, statements such as `SHOW CREATE TABLE`, and the `ST_GeometryType()` function that display data type information now display `GEOMCOLLECTION` rather than `GEOMETRYCOLLECTION`. Applications that look for `GEOMETRYCOLLECTION` should be adjusted to account for this change.

For more information, see GeometryCollection Class, and MySQL-Specific Functions That Create Geometry Values.

sys Schema Notes

- `sys` Schema objects now have a default character set of `utf8mb4`. (Bug #27675959)

Test Suite Notes

- Reduction of compiler and platform differences in GIS handling of floating-point results enables simplification of related test cases that no longer need rounding to avoid spurious test failures. Thanks to Daniel Black for the patch. (Bug #26540102, Bug #87223, Bug #27462294)

X Plugin Notes

- When using MySQL as a document store, the use of `SPATIAL` indexes on geometry-valued columns using the X DevAPI is now fully supported. Previously, a spatial index could be created, but the spatial reference system identifier (SRID) was not included in the indexed column, so the index was not used by the optimizer. The `SRID` attribute is now added to the indexed column. (Bug #27568243)
- Socket files used with X Plugin tests are now limited to 22-character file names. (Bug #27460677, Bug #89464)
- The fix for Bug #26328274 resulted in an assertion in debug builds when an empty host name was used for X Plugin connections. The issue is now fixed. (Bug #27266584, Bug #88925)
- X Plugin did not populate the password cache when a user account with no password set was used with the `PLAIN` authentication method. This caused later authentication attempts to fail. The password cache is now correctly populated with an entry if the user account's password is empty. (Bug #27257722)
- X Plugin now supports the `RESTART` SQL statement, which was added in MySQL 8.0.4 to enable a MySQL server instance to be restarted from within a client session. (Bug #27245319)
• MySQL Shell connections using X Protocol can use either TCP or Unix sockets as of version 8.0.4. This fix allows the use of encryption protocols with UNIX socket connections, so user accounts that require encrypted connections are able to connect. (Bug #27192091)

• The fix for Bug #26647488, which corrected the handling of the DATETIME data type, now also correctly decodes messages received from older versions of the plugin that do not have the new field available to identify the data type. (Bug #27169735)

• X Plugin connection attempts using the X Protocol did not return an error when the default database specified in the connection options was invalid, and the connection was allowed with a null default database. Connection attempts using the classic MySQL protocol did return an error and disallowed the connection. X Protocol connection attempts now also disallow the connection if an invalid schema is specified. (Bug #26965020)

• X Plugin is now loaded by default, meaning there is no longer any need to install the plugin.

Functionality Added or Changed

• **InnoDB**: The ibd2sdi serialized dictionary information (SDI) extraction utility for InnoDB tablespaces now supports a --pretty option that controls JSON pretty print formatting of SDI data. Pretty print formatting is enabled by default. Disabling pretty print formatting using --skip-pretty reduces the size of extracted SDI data. (Bug #27546514)

• **InnoDB**: To improve startup performance on systems with large buffer pools, buffer pool initialization is now multithreaded. (Bug #22963374, Bug #80784)

• **InnoDB**: The following redo logging optimizations were implemented:
  • User threads can now write concurrently to the log buffer without synchronizing writes.
  • User threads can now add dirty pages to the flush list in a relaxed order.
  • A dedicated log thread is now responsible for writing the log buffer to the system buffers, flushing system buffers to disk, notifying user threads about written and flushed redo, maintaining the lag required for the relaxed flush list order, and write checkpoints.
  • System variables were added for configuring the use of spin delay by user threads waiting for flushed redo:
    • innodb_log_wait_for_flush_spin_hwm: Defines the maximum average log flush time beyond which user threads no longer spin while waiting for flushed redo.
    • innodb_log_spin_cpu_abs_lwm: Defines the minimum amount of CPU usage below which user threads no longer spin while waiting for flushed redo.
    • innodb_log_spin_cpu_pct_hwm: Defines the maximum amount of CPU usage above which user threads no longer spin while waiting for flushed redo.
  • The innodb_log_buffer_size configuration option is now dynamic, which permits resizing of the log buffer while the server is running.

For more information, see Optimizing InnoDB Redo Logging.

• **macOS**: The macOS installer now asks the user for the root password to use rather than generating a random one. (Bug #22608157, Bug #80144)
• **macOS:** The MySQL preference pane included with the macOS installer now includes a setting that specifies which `my.cnf` file the server should read when started by `launchd`. (Bug #19775549, Bug #74214)

• **Solaris:** On Solaris, MySQL can now be built with Developer Studio 12.6. (Bug #27055190, Bug #88316, Bug #28165246, Bug #91214)

Logging during data directory initialization (for example, using `mysqld --initialize`) has changed:

• The startup message indicates that the server is running in initialization mode.

• A message is written indicating when initialization has completed.

• The message about the CA certificate being self-signed is suppressed.

(Bug #27557952, Bug #89693)

• The `-log` suffix is no longer appended to the server version string if logging is enabled. This affects, for example, the value of the `version` system variable. (Bug #27532097, Bug #89626)

• Server shutdown lines written to the error log now include the server version. (Bug #27522788, Bug #87910)

• `SHOW CREATE TABLE` normally does not show the `ROW_FORMAT` table option if the row format is the default format. This can cause problems during table import and export operations for transportable tablespaces. MySQL now supports a `show_create_table_verbosity` system variable that, when enabled, causes `SHOW CREATE TABLE` to display `ROW_FORMAT` regardless of whether it is the default format. (Bug #27516741)

• Performance Schema memory instrumentation overhead was reduced. (Bug #27500610)

• During data directory initialization, the server created objects using `sql_mode='''`. Now the server uses the default `sql_mode` value (which, among other settings, includes strict mode). (Bug #27476029, Bug #89495)

• The collations defined in `mysql-test/std_data/Index.xml` are for test purposes only and should not be used otherwise. Comments were added to the file to make this clear because some users were using them in non-test environments. (Bug #27421838)

• `RESET PERSIST` no longer performs an implicit commit. (Bug #27374791)

• For a persisted variable, the `SET_TIME` column of the Performance Schema `variables_info` table now reflects the time at which the variable was persisted, not server startup time. (Bug #25677422)

• Previously, `SHOW CREATE TABLE` did not show the collation for a column if the collation was the same as the table default, even if the collation was explicitly specified at table-creation time. Now, `SHOW CREATE TABLE` always shows the column collation if the collation was explicitly specified, even if the collation is the same as the table default. (Bug #11754608, Bug #46239)

### Bugs Fixed

**Important Change:** When `NO_ZERO_DATE` SQL mode is enabled, `ALTER TABLE` rejects attempts to add `DATE` or `DATETIME` columns that are declared `NOT NULL` to a nonempty table if no `DEFAULT` is specified for that column. Since a generated column has no default value, this restriction also prevented addition of a generated `DATE NOT NULL` or `DATETIME NOT NULL` column to a nonempty table. Since the expression used to generate the column can be used in place of the missing default for determining the column value, this restriction is now lifted for generated columns. (Bug #27252354)

Bugs Fixed
• **Important Change:** For `ANALYZE TABLE ... UPDATE HISTOGRAM` statements, the `WITH N BUCKETS` clause is now optional. If not specified, the default value for `BUCKETS` is 100. (Bug #89909, Bug #27637968)

• **InnoDB:** The server was stopped before a fatal error message was written to the error log. (Bug #27626478)

• **InnoDB:** When determining if a wait lock could be granted, locks held by victim transactions that were being rolled back were ignored. (Bug #27607235)

• **InnoDB:** An error occurred in `Sdi_Decompressor::decompress()` when compiling a debug build using GCC 7.2.0. (Bug #27565889)

• **InnoDB:** An assertion was raised when registering buffer pool chunks during parallel creation of multiple buffer pools. A mutex is now used to protect registration of buffer pool chunks in the buffer pool chunk map. (Bug #27565865)

• **InnoDB:** Enabling the standard monitor caused an error. (Bug #27534041)

• **InnoDB:** An internal field used to mark a column as hidden was changed from a boolean field to an enum field to support additional levels of column visibility. (Bug #27499518, Bug #89553)

• **InnoDB:** An invalid zlib-related Valgrind failure reported that a conditional jump or move depends on uninitialized value. (Bug #27482740, Bug #89509)

• **InnoDB:** An incorrect `GROUP BY` result was returned when using the `TempTable` storage engine and a `NO PAD` collation. (Bug #27454796)

• **InnoDB:** An incorrect compression length value in a page compression function caused hole punching to be skipped the first time pages are compressed. (Bug #27399897)

• **InnoDB:** Starting the server with an incorrect `--log-bin` option value raised an assertion, and subsequent attempts to start the server with a correct `--log-bin` value or without the `--log-bin` option continued to produce assertion failures. (Bug #27393317)

• **InnoDB:** A large object (LOB) reference was not updated when a LOB was inserted during an `UPDATE` operation. (Bug #27368662)

• **InnoDB:** After restarting the server, a `TRUNCATE TABLE` operation raised an assertion due to table flags that were inconsistent with remote data directory information stored in memory. (Bug #27367639)

• **InnoDB:** Attempting to create a temporary table in a file-per-table tablespace using `CREATE TEMPORARY TABLE ... TABLESPACE` syntax failed to report an error. Temporary tablespaces are only permitted in the temporary tablespace. (Bug #27361662)

• **InnoDB:** A latch taken on the first page of a large object (LOB) was not released before another thread attempted to allocate the page, causing a deadlock. (Bug #27360492)

• **InnoDB:** Converting an implicit lock to an explicit lock was optimized by the introduction of a member variable that tracks the smallest active transaction ID. Thanks to Zhai Weixiang for the patch. (Bug #27353468, Bug #89127)

• **InnoDB:** The `mysql.innodb_ddl_log` and `mysql.innodb_dynamic_metadata` tables, which are used internally by the InnoDB storage engine, are now protected. DDL and DML operations on these tables are no longer permitted, and the tables no longer appear in `INFORMATION_SCHEMA` query results. (Bug #27353455, Bug #89129)
MySQL 8.0 Release Notes

- **InnoDB**: Upgrading a MySQL 5.7 instance with an `innodb_page_size` setting of 4K caused `mysql_upgrade` to fail with a “Specified key was too long; max key length is 768 bytes” error when creating or modifying system tables. The error was suppressed to permit the `mysql_upgrade` client to run on an instance with an `innodb_page_size` setting of 4K. (Bug #27338431)

- **InnoDB**: The recovery process could not properly determine space_id values, causing the server to exit. (Bug #27331936)

- **InnoDB**: An `ALTER TABLE` operation that modified the nullability of an indexed column raised an assertion. InnoDB incorrectly assumed that a new primary key must be added instead of created from an existing unique key. (Bug #27330454, Bug #89087)

  References: This issue is a regression of: Bug #24397406.

- **InnoDB**: After an `ALTER TABLE ... IMPORT TABLESPACE` operation, metadata could not be retrieved correctly from the `INFORMATION_SCHEMA.INNODB_COLUMNS` table. (Bug #27322099)

- **InnoDB**: An `INSERT` operation involving a large object (LOB) value raised an assertion. After splitting a LOB fragment, there was insufficient space for a directory entry. (Bug #27320977)

- **InnoDB**: On a server using the utf8mb4 character set, an `INSERT` or `UPDATE` during an `ALTER TABLE` operation that changed the row format of a table that included a virtual column caused an index corruption. (Bug #27319084)

- **InnoDB**: The data retrieved from `INFORMATION_SCHEMA.INNODB_COLUMNS` was incorrect for tables containing a virtual column. (Bug #27316037)

- **InnoDB**: The server failed to start after recovery due to an invalid keyring file. The keyring file created from backup was empty. (Bug #27307740)

- **InnoDB**: A deadlock between background threads, one attempting to evict a full-text search table from the cache, and the other attempting to synchronize a table, caused InnoDB Cluster nodes to fail. (Bug #27304661)

- **InnoDB**: A typo was corrected in an InnoDB recovery message. Thanks to Daniël van Eeden for the patch. (Bug #27287979, Bug #27010613, Bug #88185)

- **InnoDB**: A message stating that encryption algorithm support is missing was reported during InnoDB recovery. (Bug #27286020)

- **InnoDB**: A missing or mismatched keyring file caused an assertion failure at startup. (Bug #27278324)

- **InnoDB**: An error was returned indicating that encryption support was missing when checking the space ID of an encrypted tablespace file. (Bug #27278100)

- **InnoDB**: A debug message that prints a “too many open files” warning is now only printed if the number of open files reaches a limit of 300. (Bug #27277844)

- **InnoDB**: Failure to skip predicate locks when releasing gaps locks raised debug assertions, as did failure to remove the supremum record bit prior releasing gaps locks on the supremum. (Bug #27272806, Bug #27294066)

- **InnoDB**: A tablespace ID discovery failure occurred when starting the server on a cloned data directory. (Bug #27265874)

- **InnoDB**: An unprotected global variable (`fil_n_file_opened`) caused an assertion when shutting down the server. (Bug #27265473)
• **InnoDB**: An assertion that is no longer valid was raised when checking for pending I/O on the file in use. (Bug #27263206)

• **InnoDB**: An empty general tablespace caused a failure when upgrading to MySQL 8.0. The empty data file was expected to have a specific number of pages. (Bug #27261822)

• **InnoDB**: A startup failure occurred because a directory specified by the `--innodb-directories` option contained a keyring data file. The `--innodb-directories` option should only recognize InnoDB tablespace files. (Bug #27253116)

• **InnoDB**: A function that calculates the increase in minimum bounding rectangle (MBR) size incorrectly returned 0, resulting in R-tree corruption. (Bug #27237472)

• **InnoDB**: A `REPLACE` operation on a temporary table raised an assertion. (Bug #27225649, Bug #27229072)

• **InnoDB**: An online `ALTER TABLE` operation on a table accompanied by concurrent DML on the same table raised an assertion. An end-of-log check was not performed prior to accessing the DML log to determine the length of a virtual column. (Bug #27158030)

• **InnoDB**: When the addition of a virtual index failed, the virtual index that was freed was not removed from the lists of virtual column indexes. (Bug #27141613)

• **InnoDB**: Adding a virtual column and index in the same statement caused an error. (Bug #27122803)

• **InnoDB**: An undo tablespace truncate operation raised an assertion. The check that determines if undo tablespace segments are inactive was invalid. (Bug #27114068)

• **InnoDB**: When the TempTable storage engine was unable to allocate memory using temporary files and mmap, an invalid error was reported indicating that the defined on-disk storage engine would be used instead. (Bug #26990375)

• **InnoDB**: Latch order was incorrectly defined in debug instrumentation causing a latch order violation during an `INSERT` operation. (Bug #26983412)

• **InnoDB**: A tablespace import operation on a server with a default row format of `REDUNDANT` raised an assertion failure. (Bug #26960215)

• **InnoDB**: A stored field based on a generated column permitted the base column to have a NULL value. (Bug #26958695)

• **InnoDB**: Evaluation of a subquery in a resolving function raised an assertion. (Bug #26909960)

• **InnoDB**: Locks related to data dictionary operations caused degradation in `CREATE TABLE` performance. (Bug #26848711, Bug #87827)

• **InnoDB**: A `LOCK INSTANCE FOR BACKUP` operation was able to acquire a backup lock without waiting for a concurrent DDL operation to finish. (Bug #26831155, Bug #87812)

• **InnoDB**: An incorrectly specified `innodb_data_file_path` or `innodb_temp_data_file_path` value returned a syntax error that did not specify the name of the system variable that caused the initialization failure. (Bug #26805833)

• **InnoDB**: An handler that was opened for a `TRUNCATE TABLE` operation caused the operation to assert. The server now closes open handlers before dropping the table.

The patch for this bug also improves `TRUNCATE TABLE` error messages related to foreign key constraints, and makes it possible to truncate a corrupted table. (Bug #26710839)
MySQL 8.0 Release Notes

- **InnoDB**: An online DDL operation that rebuilds the table raised an assertion when the last insert log record to be applied was split across two pages. (Bug #26696448, Bug #87532)

- **InnoDB**: An assertion was raised when attempting to generate a virtual column value dependent on an off-page large object value that was not yet stored. (Bug #26625652)

- **InnoDB**: An R-tree search counted delete-marked records, causing it to find more records than present in the clustered index. As result, the valid R-tree index was marked as corrupted. (Bug #26381207)

- **InnoDB**: A `RENAME TABLE` operation that renamed the schema failed to rename full-text search common auxiliary tables that were left behind when the full-text search index was removed previously, resulting in an assertion failure when attempting to drop the old schema. (Bug #26334149)

- **InnoDB**: An assertion was raised when a thread attempted to read a record containing BLOB data while another thread was writing the same data to external pages. (Bug #26300119)

  References: This issue is a regression of: Bug #23481444.

- **InnoDB**: Column prefixes for externally stored columns exceeded the amount of available memory in the online log. (Bug #25928471)

- **InnoDB**: A problematic lock that was introduced to protect writing of dynamic metadata changes to the redo log during a checkpoint was removed. A mutex is now used instead. The checkpoint LSN was also adjusted to protect dynamic metadata changes. (Bug #25834374)

- **InnoDB**: InnoDB failed to account for a virtual column when using the column offset to search an index for an auto-increment column. (Bug #25076416)

- **InnoDB**: An invalid debug condition caused a buffer pool chunk allocation failure, which resulted in an assertion failure when a purge thread attempted to access an unallocated chunk. (Bug #23593654)

  References: This issue is a regression of: Bug #21348684.

- **InnoDB**: Threads were prevented from entering the InnoDB storage engine due to ticket acquisition deadlocks. (Bug #23476050)

- **Partitioning**: An error returned by the storage engine while executing `ALTER TABLE EXCHANGE PARTITION` was not handled correctly by the server. (Bug #27320682)

- **Replication**: When MySQL was started with `--plugin-load='group_replication.so'` but Group Replication was not started, starting an asynchronous slave channel resulted in an unresponsive server. (Bug #27584141)

- **Replication**: Empty XA transactions that have been rolled back should not be recorded in the binary log or added to the set of transactions in the `gtid_executed` system variable. However, if rolled-back empty XA transactions were executed after an empty XA transaction ending with an `XA COMMIT ... ONE PHASE` statement, they were being incorrectly added to the binary log as an `XA ROLLBACK` statement, and also to the `gtid_executed` system variable. The issue is now fixed. (Bug #27435974)

- **Replication**: The Group Replication files have been moved from the `rapid` directory to the `plugin` directory. (Bug #27413789)

- **Replication**: When the system variable `gtid_next` was set to AUTOMATIC, empty XA transactions that were rolled back and not recorded in the binary log, were being incorrectly added to the set of transactions in the `gtid_executed` system variable, causing an inconsistency. The rolled-back empty XA transactions are no longer recorded in `gtid_executed`. (Bug #27407670)

- **Replication**: Group Replication conflict detection uses schema and table names as part of the Primary Key Equivalent (PKE) in order to detect and disallow conflicting transactions. The value of the
lower_case_table_names system variable changes how schema and table names are stored and externalized, which depending on the configured value could persist a table named T1 as t1. Such a difference in a group could cause inconsistencies. Now, members must all have the same value for lower_case_table_names. (Bug #27401817)

- **Replication**: Using an IP address or hostname in any Group Replication related configuration on macOS was failing. (Bug #27376511, Bug #89123, Bug #27604471)

- **Replication**: Changing the Group Replication required settings incorrectly on online secondary members could result in an unexpected halt. (Bug #27317478, Bug #27157202)

- **Replication**: The value of LAST_CONFLICT_FREE_TRANSACTION in the replication_group_member_stats Performance Schema table was not correctly synchronized across group members. (Bug #27317431)

- **Replication**: After issuing START_GROUP_REPLICATION the gtid_mode system variable is locked to prevent any modification to its value until the group is online. Any attempt to modify gtid_mode during this time is blocked. As part of the process of starting Group Replication the server needs to set super_read_only=off, which has dependencies on locks acquired by SET_GTID_MODE. This could result in Group Replication hanging and there was no possibility to connect to the server to resolve the situation. To prevent this situation, when it is not possible to acquire the locks needed by SET_GTID_MODE the operation aborts. (Bug #27312241)

- **Replication**: When a member is joining a group there is a chance of the request to join being rejected. If the rejection resulted in a retry, for example because the seed member being contacted was not in the group, then there was a possibility of the retry cycle continuing infinitely. (Bug #27294009)

- **Replication**: In the unlikely event of the group failing to start, for example because no seed member could be contacted, there is a one minute period where the group retries the start process. Attempting to modify a Group Replication plugin option during this period could result in an unexpected halt. Now, attempting to modify a Group Replication plugin option during plugin start or stop results in error. (Bug #27275312)

- **Replication**: The default binary log expiration period of 30 days is now set as the default for the binlog_expire_logs_seconds system variable, instead of the deprecated expire_logs_days system variable. binlog_expire_logs_seconds now defaults to a value of 2592000 (30*24*60*60 seconds), and expire_logs_days now defaults to zero.

In MySQL 8.0.3 and 8.0.4, the effects of binlog_expire_logs_seconds and expire_logs_days were cumulative. From MySQL 8.0.11, this is no longer the case, and the new behavior is as follows:

- If you specify a non-zero value for both variables, only the value of binlog_expire_logs_seconds is used to determine the binary log expiration period, and the value of expire_logs_days is ignored.

- If you specify a non-zero value for only one variable, that value is used to determine the binary log expiration period, and the default for the other is not applied.

- If you specify a value for neither of the variables, the default of 30 days from binlog_expire_logs_seconds is used as the binary log expiration period.

An error message is issued if you attempt to set one of the variables dynamically when the other is already set, and a warning message is issued if you specify a non-zero value for both variables at startup. A deprecation warning is also issued if you set a value for expire_logs_days.

To disable automatic purging of the binary log, specify a value of 0 explicitly for binlog_expire_logs_seconds, and do not specify a value for expire_logs_days. For compatibility with earlier releases, automatic purging is also disabled if you specify a value of 0 explicitly.
for `expire_logs_days` and do not specify a value for `binlog_expire_logs_seconds`. In that case, the default for `binlog_expire_logs_seconds` is not applied. You no longer need to set both variables to zero to disable automatic purging. (Bug #27252658)

**Replication:** Attempting to uninstall the Group Replication plugin on a server configured to start the plugin automatically at boot could result in a deadlock. Now, the uninstall process checks to ensure that the plugin is not in the process of starting and fails gracefully if it is. (Bug #27179336)

**Replication:** The DDL statements `CREATE DATABASE`, `ALTER DATABASE`, and `DROP DATABASE` are no longer allowed to execute in parallel with other statements on a multithreaded slave. These statements require an exclusive metadata lock on the schema, causing a theoretical possibility of a deadlock when `slave_preserve_commit_order=1` is set on the replication slave. (Bug #27165446)

**Replication:** On a replication slave with binary logging disabled, an empty DDL statement (for example, due to a replication filter) raised an assertion. With binary logging enabled, `BEGIN` and `COMMIT` statements would be written to the binary log to produce a transaction, but these statements were not available with binary logging disabled, causing the assertion. MySQL Server now checks for the situation where the DDL transaction is empty and is not marked as started, and handles it without raising the assertion. (Bug #27164661)

**Replication:** A new system variable `rpl_read_size` is available to set the minimum amount of data in bytes that is read from the binary log files and relay log files. The default read size is 8KB. If heavy disk I/O activity for these files is impeding performance for the database, increasing the read size might reduce file reads and I/O stalls if the files are not being cached by the operating system.

Note that a buffer the size of this value is allocated for each thread that reads from the binary log and relay log files, including dump threads on masters and coordinator threads on slaves. Setting a large value might therefore have an impact on memory consumption for servers.

Thanks to the Facebook team for the patch. (Bug #27147095)

**Replication:** On a server with GTIDs enabled, if `RESET MASTER` was used when a transaction had been flushed but not yet committed, the server GTIDs were left in an incorrect state. At this point, the GTID for the transaction had been added to the set of owned GTIDs, but not yet to the set of executed GTIDs. The GTID was removed from the binary log when the content of the uncommitted transaction was cleared by the `RESET MASTER` statement, but the GTID could not be reused because it was already owned, so the sequence of GTIDs had a gap.

With this fix, `RESET MASTER` acquires a global read lock to ensure that no transactions are in commit stage during its operation, and no new transactions enter into commit stage during its operation. If there are any ongoing commits, `RESET MASTER` waits until they are complete. The global read lock is released when the reset is finished, and then storage engines are notified of the successful reset. If `FLUSH TABLES WITH READ LOCK` has already been executed for the thread, `RESET MASTER` does not retry the lock. (Bug #27041759)

**Replication:** Attempts to set a GTID for an unsafe or empty XA transaction using a `SET @@SESSION.GTID_NEXT` statement were not being handled correctly. On a server with `gtid_mode` set to `OFF` or `OFF_PERMISSIVE`, and `enforce_gtid_consistency` set to `OFF`, setting a GTID for an unsafe XA transaction (creating or dropping a temporary table) caused an inconsistency in the server’s GTID violation counter and raised an assertion. Testing also showed that setting a GTID for an empty XA transaction caused an incorrect sequence of events to be written to the binary log. Both these situations are now handled correctly. (Bug #27041402, Bug #88272)

**Replication:** The error message issued for a server stop while rotating the binary log did not state the correct cause for the issue. The error message has been enhanced to include the actual error causing the server stop as part of the message. A new error `ER_OOM_SAVE_GTIDS` has also been added for the
situation where an out-of-memory error occurred while saving the set of GTIDs from the last binary log into the `mysql.gtid_executed` table. (Bug #27040070, Bug #88262)

- **Replication:** Group Replication uses transaction write set extraction for conflict detection on group members. A performance regression was found in this process during detailed performance analysis, which is handled by this bug fix. Memory allocation has been optimized for write set extraction and memory copy operations have been reduced. Also, foreign key write sets are now only collected when the current table has foreign keys. (Bug #27016053)

- **Replication:** The statements `CREATE USER IF EXISTS` (or `IF NOT EXISTS`) and `ALTER USER IF EXISTS` (or `IF NOT EXISTS`) were written to the binary log even when the query resulted in an error. MySQL Server now checks for errors that cause these queries to fail (for example, an invalid plugin was specified), and does not log the statement in that situation. Note that if these statements succeed but have no effect on the master because the condition is not met, the statements are written to the binary log, as the condition might be met on a replication slave (see Bug #25813089, Bug #85733). (Bug #26680035)

  References: See also: Bug #25813089, Bug #85733.

- **Replication:** In a multi-primary group, when a member was also configured with an asynchronous replication channel, there was a possibility that the asynchronous channel could start before Group Replication started. This could result in the asynchronous channel processing transactions before the member became an online member of the group, causing issues when members tried to join the group. The fix ensures that asynchronous channels on group members do not start until the member has become online. (Bug #26648393)

- **Replication:** Following the introduction of support for atomic DDL in MySQL 8.0, a replication slave that supports atomic DDL can diverge from a replication master at an earlier MySQL version that does not support atomic DDL. If the DDL statement can only be partly applied, the slave using MySQL 8.0 rolls back the whole DDL statement, but the master using an earlier version might commit the valid parts of the DDL statement. An error message is now logged in the event of an unsuccessful commit for an atomic DDL statement on the master and the slave, and you should check that the states of the master and the slave have not diverged before proceeding. (Bug #26133488)

- **Replication:** When GTIDs are enabled on a replication master and slave, and the slave connects to the master with the `MASTER_AUTO_POSITION=1` option set, the master must send the slave all the transactions that the slave has not already received, committed, or both. If any of the transactions that should be sent by the master have been already purged from the master's binary log, the master sends the error `ER_MASTER_HAS_PURGED_REQUIRED_GTIDS` (1789) to the slave, and replication does not start.

  The message provided for the error `ER_MASTER_HAS_PURGED_REQUIRED_GTIDS` has been changed to provide advice on the correct action in this situation, which is for the slave to replicate the missing transactions from another source, or for the slave to be replaced by a new slave created from a more recent backup. The message advises that the master's binary log expiration period can be revised to avoid the situation in future. In addition, the master now identifies the GTIDs of the purged transactions and supplies them in its error log in the warning message `ER_FOUND_MISSING_GTIDS` (11809), so that you do not need to calculate the missing GTIDs manually. (Bug #26004541)

- **Replication:** For updates to virtual generated columns containing the BLOB data type, both the old and the new BLOB values are required by some storage engines for replication. This fix extends the same behavior to JSON and GEOMETRY data types, which are based on the BLOB data type and so produce the same issue when the old value is not stored. (Bug #25873029)

- **Replication:** A behavior change in MySQL 8.0.0 moved the identification of generated columns from before to after the calculation of the write set for binary logging. This caused an issue with NDB cluster
replication, because that storage engine omits the generated columns from the log event if they are unnecessary, leading to NULL values being set for them by the applier thread on the replication slave. The identification of generated columns has now been moved back to before the write set calculation, so that the issue does not occur. (Bug #25827408)

**Replication:** On a multithreaded replication slave (with slave_parallel_workers greater than 0), the slave’s lag behind the master was not being reported by the Seconds_Behind_Master field for SHOW SLAVE STATUS. The value is now reported correctly. Thanks to Robert Golebiowski for the patch. (Bug #25407335, Bug #84415)

**Replication:** A plus sign (+) can be used with a GTID set in the statement SET @@GLOBAL.GTID_PURGED to indicate that the specified GTID set does not intersect with gtid_executed. The plus sign was also being permitted for GTID sets specified for the functions GTID_SUBSET() and GTID_SUBTRACT(), where it was not meaningful. The plus sign can no longer be specified with these functions. Also, error messages relating to the SET @@GLOBAL.GTID_PURGED operation have been improved. (Bug #24675979)

**Replication:** When invoked with the options --read-from-remote-server and --hexdump, mysqlbinlog was not able to produce a hex dump of the binary log contents following an SQL statement that generated an autoincrement value, referenced a user-defined variable, or invoked RAND(). The event types for these events are followed by an informational row query log event, and mysqlbinlog caches the original event for printing when the subsequent row query log event is received. The pointer to the memory containing the original event was invalidated when the subsequent event was received, so the original data could not be accessed to produce the hex dump. The issue has now been fixed. (Bug #24674276)

**Replication:** A number of changes were made to the binary log decoding procedure to improve handling of invalid or corrupted binary log entries. (Bug #24365972)

**Replication:** Following the introduction of binary logging for XA transactions WL#6860, an assertion could be raised in debug builds during replication from a master with the feature to a slave without the feature, if MASTER_AUTO_POSITION=1 was set on the slave. The assertion has been removed, so that debug builds now have the same behavior as non-debug builds, and can attempt replication of unsupported event types whether or not MASTER_AUTO_POSITION=1 is set. (Bug #20677683)

**Replication:** When using group_replication_ip_whitelist, it was possible to configure a group so that it functioned even though all members could not establish the internal group communication connection to each other, resulting in inconsistent behavior. Now, incoming connections are accepted if the IP is in the white list or if the IP belongs to a current member of the XCom configuration. This ensures members are always able to create the internal network required for group communication. (Bug #87834, Bug #26846549, Bug #27406775)

**Linux:** Builds on Alpine Linux now take advantage of AIO support. (Bug #27327874)

**macOS:** On macOS, mysql_config --libs did not list any SSL libraries. (Bug #27232576)

**Microsoft Windows:** On Windows, the RESTART statement is implemented by having mysqld fork, with one process acting as a monitor to the other, which acts as the server. This makes determining the server process to attach to for debugging more difficult. To alleviate this, starting the server with --gdb now suppresses forking. A side effect is that for a server started with this option, RESTART simply exits and does not restart. (Bug #27273229)

**JSON:** The JSON_QUOTE() function could in some situations write the quoted string to the same buffer that holds the input string, which could lead to wrong results. Now a check is made to be sure that it does not attempt write into the same buffer that it is reading from. (Bug #27312444)
MySQL 8.0 Release Notes

• **JSON**: `JSON_SEARCH()` no longer performs any modification of cached `Json_path` objects, now making any needed updates to a `String` object that represents the path instead. This saves on the number of round trips required between the path's `Json_path` and `String` representations, which speeds up execution. In addition, the `one_or_all` and `escape_char` arguments to `JSON_SEARCH()` were unnecessarily cached; these arguments are no longer cached, which should also improve this function's execution time. (Bug #87383, Bug #26614455)

References: See also: Bug #21450084, Bug #77785, Bug #21472872.

• A check was implemented based on data dictionary version information to prevent starting the MySQL 8.0 server with a data directory created by a later MySQL release. The check was necessary to prevent an in-place downgrade, which is currently not supported. (Bug #27708249)

• Protected data dictionary tables are no longer accessible by statements placed in a server initialization file. (Bug #27674311, Bug #90010)

• Data dictionary and `INFORMATION_SCHEMA` version numbers were synchronized with the new MySQL server version number. (Bug #27674285)

• The server did not handle correctly certain `LIKE` queries using a `BLOB` column with a prefix index. (Bug #27660560)

• For builds configured using `--with-protobuf=SYSTEM`, building failed on systems with Protobuf 3.5 installed. (Bug #27638713, Bug #89915)

• The microseconds part of timestamps stored with persisted variables in `mysql-auto.cnf` was always 000000. (Bug #27629719)

• Debian packages now handle dependencies for MeCab plugin dictionary files. Fedora packages now handle dependencies for both MeCab packages, not just the utf-8 package. (Bug #27612730, Bug #27613539)

• The change in MySQL 8.0.11 for platforms that use systemd to run `mysqld` as a normal process (`Type=notify`) and communicate using a socket file did not work on SLES before SLES 12.2. (Bug #27604999)

• The `NON_UNIQUE` column in the `INFORMATION_SCHEMA.STATISTICS` table had type `BIGINT` prior to MySQL 8.0, but became `VARCHAR` in MySQL 8.0 with the introduction of the data dictionary. The `NON_UNIQUE` column now has an integer type again (`INT` because the column need not be as large as `BIGINT`). (Bug #27593348, Bug #89793)

• With some inputs, `ST_Crosses()` could cause a server exit. (Bug #27576700)

• Some error messages were being written to the error log with an error ID of 0 rather than a legitimate error ID. (Bug #27575608)

• `mysql_upgrade` did not set `MYSQL_SERVER_PUBLIC_KEY` correctly, leading to failure to read the key file. (Bug #27568278)

• If `autocommit` was disabled, `mysql_upgrade` produced an error when upgrading from MySQL 5.7 to 8.0. (Bug #27549249)

• For debug builds, displaying very long stage names from the `PROCESSLIST_STATE` column of the Performance Schema `threads` table could raise an assertion. (Bug #27545688)

• Group Replication failed to start if certain required variables had been set as persisted variables, or if persisted variables were not set in proper timestamp order. (Bug #27545544, Bug #27522405)
MySQL 8.0 Release Notes

- `gtid_purged` handling had a memory leak. (Bug #27537968, Bug #89645)

- Using `SET_PERSIST` to persist `optimizer_trace_offset` resulted in an incorrect variable value after server restart. (Bug #27536421)

- The `validate_password` component could leak memory. (Bug #27521770, Bug #89597)

- A typo in `sql/handler.h` was corrected. Thanks to Su Tristan for the patch. (Bug #27516280, Bug #89594)

- The Performance Schema `variables_info` table displayed incorrect `VARIABLE_SOURCE` and `VARIABLE_PATH` values for variables set within option files specified by `!include` or `!includedir` directives. (Bug #27514223)

  References: This issue is a regression of: Bug #25563891.

- After upgrading from MySQL 5.7 to MySQL 8.0, the first attempt to start the server on a MySQL 5.7 data directory with `--default-time-zone` set to a specific time value such as "+00:00" caused the server to exit. (Bug #27512609, Bug #89584)

- Lookups of character set internal numbers was not thread safe and could cause memory leaks. (Bug #27507950)

- Using the `-DWITH_LIBEVENT=system` and `-DWITH_ICU=system` CMake options together caused configuration to fail. (Bug #27505535, Bug #89398)

- Some messages were being written to the server error log using client error IDs. (Bug #27503787, Bug #89562)

- When run in key migration mode, the server could report an error for successful operations. (Bug #27493997)

- Upgrades from MariaDB to MySQL Community Edition failed on Fedora 27. (Bug #27484835)

- `ALTER TABLE` could hang in a Waiting for tablespace metadata lock state. (Bug #27472087, Bug #89487)

- Selecting from the Performance Schema `status_by_thread` or `variables_by_thread` table was not thread safe and could yield incorrect results. (Bug #27471510)

- `INSERT ... ON DUPLICATE KEY UPDATE` could be handled improperly if a source table produced no rows. (Bug #27460607)

- `SET_PERSIST_ONLY` ignored `--skip-grant-tables` and continued to perform checking against the dynamic privileges needed to set system variables. (Bug #2745943)

- For RPM packages and Docker RPM packages, the included `my.cnf` file now includes information indicating how to revert to the previous default authentication plugin (changing `caching_sha2_password` to `mysql_native_plugin`), for compatibility with older clients. (Bug #27454015, Bug #27675380)

- In string comparisons using `>`, only one operand was handled as `utf32`. (Bug #27452148)

- A server exit during InnoDB initialization caused AddressSanitizer (ASan) to report a memory leak. The patch also implements proper handling of the data dictionary properties table in cases where the table contains data that cannot be parsed. (Bug #27447981, Bug #89433)

- With `-DWITH_SSL=system`, if CMake could not find the system OpenSSL libraries and header files, it produced confusing messages. Now it exits immediately with an error. (Bug #27447874)
MySQL 8.0 Release Notes

- Configuring with `-DWITH_INNODB_EXTRA_DEBUG=ON` resulted in a linker error. (Bug #27444255, Bug #89412)

- Configuring with `-DWITH_ZLIB=system` resulted in a linker error. (Bug #27435371, Bug #89373)

- Metadata from result sets for `UNION ALL` queries could say `NEWDATE` rather than `DATE`. (Bug #27422376)

- In builds with Undefined Behavior Sanitizer enabled, negation of -92237203685477580 could cause a server exit. (Bug #27419181)

- With protocol compression enabled, a spurious assertion could be raised. (Bug #27418207, Bug #89324)

  References: This issue is a regression of: Bug #17922198.

- Performance related to fetching tablespace objects from the data dictionary to populate the data dictionary cache was improved. (Bug #27402243)

- If the server was started with `--skip-grant-tables`, clients that authenticate with the `caching_sha2_password` plugin were unable to connect. (Bug #27400095, Bug #89267)

- Linux RPM and Debian packages now include dependency information for the Perl JSON module required to run the MySQL test suite. Linux RPM packages now include dependency information for the Perl Digest module required to run the MySQL test suite. (Bug #27392800, Bug #89250, Bug #27392808, Bug #89244)

- A new system variable, `default_collation_for_utf8mb4`, is provided for internal use by replication. The system variable is set to the default collation for the `utf8mb4` character set. The value of the variable is replicated from a master to a slave so that the slave can correctly process data originating from a master with a different default collation for `utf8mb4`. The variable is primarily intended to support replication from a MySQL 5.7 or older master server to a MySQL 8.0 slave server, or group replication with a MySQL 5.7 primary node and one or more MySQL 8.0 secondaries. The default collation for `utf8mb4` in MySQL 5.7 is `utf8mb4_general_ci`, but it is `utf8mb4_0900_ai_ci` in MySQL 8.0. The variable is not present in releases earlier than MySQL 8.0, so if the slave does not receive a value for the variable, it assumes the master is from an earlier release, and sets the value to the previous default collation `utf8mb4_general_ci`.

  The default `utf8mb4` collation is used in the following statements:

  - `SHOW COLLATION` and `SHOW CHARACTER SET`.

  - `CREATE TABLE` and `ALTER TABLE` having a `CHARACTER SET utf8mb4` clause without a `COLLATION` clause, either for the table character set or for a column character set.

  - `CREATE DATABASE` and `ALTER DATABASE` having a `CHARACTER SET utf8mb4` clause without a `COLLATION` clause.

  - Any statement containing a string literal of the form `'_utf8mb4'some text'_utf8mb4'` without a `COLLATE` clause.

    (Bug #27389878, Bug #27081073)

  - `DESCRIBE` with a column name option ignored the column name. (Bug #27387773, Bug #89224)

  - When run in key migration mode, the server ignored invalid options. (Bug #27387331)

  - `SHOW GRANTS` for an anonymous user could cause a server exit. (Bug #27375069)
- Path name normalization could fail for very long path names. (Bug #27368298)
- During configuration, CMake assumed that rpcgen was available rather than checking for it. (Bug #27368078)
- The client authentication process could use memory after it had been freed. (Bug #27366143)
- -DWITH_ZLIB=system could cause other CMake feature tests to fail. (Bug #27356658, Bug #89135)
- On some systems, using mysql --initialize to initialize the data directory was very slow. (Bug #27349579, Bug #89122)
- Builds using RPM source packages now use a secure connection if Boost must be downloaded. (Bug #27343289, Bug #89104)
- An internal check_datetime_range function that converted an ULLONGLONG time value raised an assertion on Windows. (Bug #27340709)
- Too much locking was being done for the caching_sha2_password authentication plugin (one lock/unlock per connection). This is now improved to one lock/unlock per plugin install/uninstall. (Bug #27335346)
- A CREATE TABLE ... LIKE operation that implicitly assigned 'innodb_system' as the tablespace name for table partitions raised an assertion. (Bug #27331588)
- SET PERSIST_ONLY could store large-valued variables incorrectly. (Bug #27322254)
- The audit_log plugin could write statements to the binary log even with binary logging disabled. (Bug #27315321)
- Certain queries using GREATEST() or LEAST() produced heap-use-after-free errors. (Bug #27312703)
- An external_language column was added to the mysql.routines data dictionary table to support stored routines in different languages. The data stored in this column can be accessed through the EXTERNAL_LANGUAGES column of the INFORMATION_SCHEMA.ROUTINES table. (Bug #27309116, Bug #89038)
- It is now prohibited to start the server with a lower_case_table_names setting that is different from the setting used when the server was initialized. The restriction is necessary because collations used by data dictionary table fields are based on the setting defined when the server is initialized, and restarting the server with a different setting would introduce inconsistencies with respect to how identifiers are ordered and compared. (Bug #27309094, Bug #89035)
- For accounts that authenticated using the auth_sock authentication plugin, the server was unable to accept connections from clients from older MySQL versions. (Bug #27306178)
- An audit_log plugin memory leak was corrected. (Bug #27302151)
- The INFORMATION_SCHEMA.FILES table now obtains from storage engines the information needed for the following columns: LOGFILE_GROUP_NAME, LOGFILE_GROUP_NUMBER, ROW_FORMAT, VERSION. (Bug #27292586)
- Dropping a user-defined function did not always remove its entry from the Performance Schema user_defined_functions table. (Bug #27270498)
- To reduce its size and storage footprint, serialized dictionary information (SDI) is now generated in a compact JSON format. (Bug #27265584)
- Concurrent execution of RESET PERSIST and SET PERSIST from multiple sessions could cause a server exit. (Bug #27264789)
• Tablespace statistics could be cached even if the statistics-collection process encountered an error. (Bug #27259963)

• `SET PERSIST` could fail to find the proper directory into which to write the `mysqld-auto.cnf` file. (Bug #27253828)

• `ALTER TABLE` could not create generated `NOT NULL` geometry columns if the table contained data. (Bug #27252609)

• The Performance Schema variables_info table displayed incorrect `VARIABLE_SOURCE` values for variables that were set in `my.cnf` and also persisted to `mysqld-auto.cnf`. (Bug #27252077)

• Improper handling of plugin loading and unloading could cause a server exit. (Bug #27247280, Bug #27297704)

• The `RESTART` statement did not work for Ubuntu packages. (Bug #27245918)

• LDAP authentication plugins were not built on FreeBSD. (Bug #27238252)

• The server did not handle queries correctly when a cached value was evaluated as `NULL` by a windowing function, or when its argument was part of a `GROUP BY` with a `ROLLUP` operation which could be evaluated as `NULL`. (Bug #27233287)

• Windowing functions such as `LAST_VALUE()` did not work correctly with tables containing `NOT NULL` columns in all cases. (Bug #27230463)

• Role cache invalidation could be performed incorrectly. (Bug #27225806)

• Incorrect handling of persisted variables at server startup could result in a server exit. (Bug #27224682)

• For upgrades from MySQL 5.7 to 8.0, the `sql_mode` value for object definitions could contain `NOT_USED`. (Bug #27219709)

• `JSON_TABLE()` failed if no default database was selected. (Bug #27217897)

• Adding a unique index to an `InnoDB` table on which multiple locks were held could raise an assertion. (Bug #27216817)

• `CREATE VIEW` statements that used `JSON_TABLE()` in the view definition failed if no database was selected. (Bug #27189940)

• The Doxygen documentation for `CLIENT_SSL_VERIFY_SERVER_CERT` was corrected. Thanks to Bradley Grainger for the patch. (Bug #27189363, Bug #88686)

• Queries on Performance Schema replication tables could return incorrect results, particularly when the execution plan used an index. (Bug #27184567)

• For some statements, the `FILE` privilege was not properly checked. (Bug #27160888)

• To better enable setting persisted variables at startup in the same order they were persisted with `SET PERSIST`, settings in `mysqld-auto.cnf` are first sorted based on timestamp (which is now stored in the file). (Bug #27157520)

• Setting the `dragnet.log_error_filter_rules` system variable to a very long value could cause a server exit. (Bug #27120953)

• For platforms that use systemd (see Managing MySQL Server with systemd), systemd was not always able to infer the state of a double-forked `mysqld` process. Consequently, systemd would attempt to restart `mysqld` even when that process terminated with status 1. systemd is now configured to run `mysqld` as a normal process (`Type=notify` rather than `Type=forking`). If the name of a socket file
is specified in the environment variable `NOTIFY_SOCKET`, mysqld attempts to open a connection for communicating with systemd and writes its state changes there. (Bug #27109556, Bug #88463, Bug #26538598, Bug #87210)

- `ST_IsValid()` returned incorrect results for some geographic polygons on non-WGS 84 ellipsoids. (Bug #27074700)

- A multiple-insert statement on a table containing a `FULLTEXT` key and a `FTS_DOC_ID` column caused a server error. (Bug #27041445, Bug #88267)

References: This issue is a regression of: Bug #22679185.

- In strict SQL mode, assignment of invalid values to `AUTO_INCREMENT` columns could be handled incorrectly, resulting in an assertion being raised. (Bug #27041393, Bug #88273)

- Thread stack exhaustion could raise an assertion rather than returning an error. (Bug #27041350, Bug #88277)

- `SET PERSIST_ONLY` did not properly consider whether the runtime validation function for persisted variables should be invoked. (Bug #27016247)

- The `audit_log` plugin could mishandle aborts of event executions, causing a server exit. (Bug #27008133)

- Some boundary cases for negation of large signed integers were corrected. (Bug #27004880, Bug #27008075)

- An `ALTER TABLE` operation attempted to set the `AUTO_INCREMENT` value for table in a discarded tablespace. (Bug #26935001)

- `MyISAM` index corruption could occur for bulk-insert and table-repair operations that involve the repair-by-sorting algorithm and many (more than 450 million) rows. (Bug #26929724, Bug #88003, Bug #28483283)

- The Performance Schema could produce `DIGEST_TEXT` values with a trailing space. This no longer occurs. (Bug #26908015)

- Dropping an index from a system table could cause a server exit. (Bug #26881798)

- A prepared statement using `CREATE TABLE ... SELECT` led to unexpected behavior when it referred in a `GROUP BY` to a view having the same name. (Bug #26881703)

- With auto-commit disabled and an `XA` transaction in PREPARED state, attempts to execute `XA COMMIT` or `XA ROLLBACK` failed. (Bug #26848877, Bug #87836)

- Some diagnostic messages produced by LDAP authentication plugins misleadingly suggested an error when no error had occurred. (Bug #26844713)

- Initialization code for `mysql_upgrade` and `mysqlpump` was reorganized to avoid assertion failures. (Bug #26802211)

- Full-text searches could raise an assertion due to improper handling of errors that occurred while attempting to acquire metadata locks. (Bug #26799898)

- A keyring file created by the `keyring_file` plugin on a 32-bit server was inaccessible by the `keyring_file` on a 64-bit server, and vice versa. (Bug #26793060)

- Metadata locks for column statistics were not displayed properly in the Performance Schema `metadata_locks` and `events_waits_` tables. (Bug #26772858, Bug #87708)
• The `my_snprintf` plugin service was removed and reimplemented using C++11 `snprintf`. (Bug #26696147, Bug #87547)

• After an in-place upgrade from MySQL 5.7 to MySQL 8.0 on a server with a large number of tables, server startup failed to complete due to excessive memory consumption during data dictionary creation. (Bug #26486160)

• An error occurring in a statement that modifies user privileges could result in deadlock for other transactions attempting to access the user privilege cache. (Bug #26475282)

• The thread pool plugin logged too much information for failed connections. (Bug #26368725, Bug #86863)

• Enabling multiple components concurrently could result in a server exit. (Bug #26171471, Bug #86514)

• A malformed `mysqld-auto.cnf` file could cause a server exit. (Bug #26085774)

• Creating a table in a reserved tablespace did not return an error. (Bug #26073851, Bug #86309)

• For debug builds, using `KILL` to terminate a stored routine could raise an assertion. Thanks to Laurynas Biveinis for the patch. (Bug #26040870, Bug #86260)

• If the `init_connect` system variable was set, its contents could not be executed by clients with expired passwords, who therefore were prevented from connecting. Now, if a client has an expired password, `init_connect` execution is skipped, which enables the client to connect and change password. (Bug #25968185)

• Dates using the `YYYYMMDD` format were not recognized correctly in a query meeting all three of the following conditions:
  The query performed a left join.
  A `DATE` column in the inner table of the join was part of a multi-column primary key.
  Every column in the inner table’s primary key was compared with another value; this could be either a literal or a column value. (Bug #25949639)

• An in-place `ALTER TABLE` operation on a table with foreign keys resulted in a table definition mismatch. The new table definition passed to storage engine methods during the `ALTER TABLE` execution contained invalid foreign key names. (Bug #25915132, Bug #85997)

• It was possible to assign nonexistent roles to an account as its default roles. (Bug #25755666, Bug #85561)

• Using the C API, when trying to execute an `INSERT` prepared statement with `CURSOR_TYPE_READ_ONLY` set, the client hung. (Bug #25701141, Bug #85105)

• `RENAME USER` failed even though the user or role to be renamed was not present in any role graph. (Bug #25658586)

• MySQL client programs could exit unexpectedly if malformed client/server protocol packets were received. (Bug #25471090)

• Memory statistics collected by the Performance Schema could be incorrect due to race conditions. (Bug #25212799)

• Incorrect handling by the `CONNECTION_CONTROL` plugin of an internal hash led to spurious messages in the error log and eventual server exit. (Bug #25052009)
MySQL 8.0 Release Notes

• **CURRENT_ROLE()** and **ROLES_GRAPHML()** now return a string with the **utf8** character set. Previously, they incorrectly returned a binary string. (Bug #24843257)

• Killing **INSTALL COMPONENT** or **UNINSTALL COMPONENT** could result in multiple rows for a single component in the **mysql.component** system table. (Bug #24660436)

• Performing **SHOW PROCESSLIST** while running a high load concurrently using the X Plugin could lead to an unplanned server exit. (Bug #24621170)

  References: See also: Bug #23057045.

• The regular expression implementation in MySQL uses a **String** object—intended to be initialized as empty—to hold the current subject, and used the literal "" for this value. This could interfere with other functions that do likewise. This is fixed by constructing the **String** with a **nullptr** instead. (Bug #23276471)

• **mysqlpump** could leak memory or exit when errors occurred. (Bug #23102944)

• LDML 2.8 collation definitions could cause unexpected server behavior. (Bug #22819030)

• Log-opening operations accessed log-name system variables without holding a lock on them. Multiple threads accessing such a variable could lead to a race condition and unexpected server behavior. (Bug #22591899)

• When the range optimizer computed the prefix for a string of sufficient length, it was possible for it to truncate the string in the middle of a character, which could lead to assertion and other failures in debug builds. Now steps are taken in such cases to make sure that the string is truncated at a character boundary. (Bug #22475473, Bug #13712655)

• Queries having subqueries or expressions in a **GROUP BY** clause sometimes returned random results due to reading of uninitialized data. (Bug #20035304)

  References: See also: Bug #21922202.

• Queries having subqueries or expressions in the **GROUP BY** clause could in some situations return random results, due to reading of uninitialized data. (Bug #20035304)

  References: See also: Bug #21922202.

• The **audit_log** plugin did not log placeholder values for prepared statements. (Bug #16617026)

• **ALTER TABLE ... RENAME** operations could lose foreign keys defined on the table if executed using the **COPY** algorithm. (Bug #11756183, Bug #48070)

• Integer columns in **UNION** statements could be cast to **BIGINT** even if a smaller integer type was more appropriate. (Bug #11747567, Bug #33004)

• Added **ER_REGEXP_INVALID_CAPTURE_GROUP_NAME** (Error 13110). (Bug #89796, Bug #27597980)

• Window function row-buffer handling has been refactored to reduce the number of handler reads by 25%. (Bug #89512, Bug #27484133)

• For some frames, the functions **VARIANCE(), VAR_POP(),** and **VAR_SAMP()** produced wrong results when evaluated with the **windowing_use_high_precision** system variable set to **false**. (Bug #89390, Bug #27438725)

• Generated columns with a prefix index were not considered when the optimizer attempted to substitute an expression with an equivalent generated column. This prevented use of prefix indexes to speed up queries with predicates employing a generated column expression. (Bug #89291, Bug #27403367)
• **FIRST_VALUE()** did not always return **NULL** for an empty frame. (Bug #89116, Bug #27348276)

• Regular expression functions accepting optional arguments did not return **NULL** as expected when one or more of these arguments was passed as an explicit **NULL**. (Bug #88872, Bug #27252630)

• Row-based replication used the wrong set of indexes on the slave. (Bug #88847, Bug #27244826)

• When executed with no default database having been chosen, **EXPLAIN EXTENDED** raised Error 1046 **No database selected** instead of the SQL syntax error expected due to the fact that the **EXTENDED** keyword has been removed. (Bug #88591, Bug #27153116)

• The CMMake option **OPTIMIZER_TRACE=0** had no effect on builds and so is no longer used. (Bug #88520, Bug #27130109)

• A **SELECT** statement using **GROUP BY** with no constant or aggregate column failed a check for functional dependencies, correctly, on the first attempt but succeeded, wrongly, on the next. (Bug #88474, Bug #27427677)

  References: See also: Bug #21807579.

• A **CREATE TABLE ... SELECT** statement with a **UNION** in the **SELECT** failed in strict mode for a **DATE** column declared as **NOT NULL**. (Bug #87711, Bug #27068222)

• Prepared statements using nested sub-selects were not always handled correctly. (Bug #87484, Bug #26657904)

• The cost estimates for a query containing window functions did not take into account the cost of **PARTITION BY** or **ORDER BY**, nor did they include the cost of using the frame buffer. (Bug #87373, Bug #26612356)

  References: See also: Bug #26502118.

• When processing a query with a subquery, the subquery is no longer optimized if outer query is known to produce an empty result; in such cases, **EXPLAIN** now shows **Not optimized, outer query is empty**.

This change is made because, when the table used by the outer query is found empty at optimization, this sets the join column to **NULL** even when defined as not nullable, which leads to a conflict when optimizing the subquery.

An exception is made if the outer query has aggregates without any **GROUP BY**; this means that it has a non-empty result, and that any subquery in the **SELECT** list must be evaluated and optimized, which reopens the nullability issue, which is now handled correctly for such cases. (Bug #83115, Bug #24713879)

  References: See also: Bug #83216, Bug #24760317.

• When selecting from all columns making up a unique key containing nullable columns, with all columns in the **WHERE** condition set to non-null values, MySQL did not take into account their uniqueness, with the result that **only_full_group_by** missed detecting a functionally dependent column. (Bug #79291, Bug #22279903)

• When using a partial index, the optimizer performed a more expensive table lookup instead of using the index, even when the partial index covered the entire set of data needed. (Bug #74359, Bug #19806106)

### Changes in MySQL 8.0.5 - 8.0.10 (Skipped version numbers)

There are no release notes for these skipped version numbers.
Changes in MySQL 8.0.4 (2018-01-23, Release Candidate)

Note
This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)

Beginning with MySQL 8.0.4, macOS 10.13 is a supported platform.

- Compilation Notes
- Component Notes
- Configuration Notes
- Deprecation and Removal Notes
- Function and Operator Notes
- InnoDB Notes
- Installation Notes
- Keyring Notes
- Logging Notes
- Packaging Notes
- Performance Schema Notes
- Security Notes
- Server Administration
- Spatial Data Support
- Test Suite Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed

Compilation Notes

- **Linux**: Binary packages on Linux platforms now are built using GCC 6. The optimization level has changed from `-O3` to `-O2`.

- **macOS; Microsoft Windows**: For compiling MySQL from source, the `-DWITH_SSL=system CMake` option now works on Windows and macOS. (Bug #26907731, Bug #87938)
MySQL 8.0 Release Notes

• On platforms for which the GNU gold linker is used, removal of unused functions is now enabled, reducing the size of binaries. (Bug #26612067, Bug #87372)

• #include directives in source files were rewritten and reorganized to be unambiguous. (Bug #26597243, Bug #87358, Bug #26897738)

• The BUILD directory containing compilation scripts is no longer maintained and has been removed from MySQL source trees. (Bug #26576219, Bug #87323)

• The minimum version of the Boost library for server builds is now 1.65.0. (Bug #26574924, Bug #87317)

• MySQL can now be linked against OpenSSL 1.1 on Unix and Unix-like systems. (Bug #25094892, Bug #83814)

• Work was done to clean up the source code base, including: Removing unneeded CMake checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

Component Notes

• The validate_password plugin has been reimplemented to use the server component infrastructure. To install and uninstall the validate_password component, use these statements:

```plaintext
INSTALL COMPONENT 'file://component_validate_password';
UNINSTALL COMPONENT 'file://component_validate_password';
```

INSTALL PLUGIN and UNINSTALL PLUGIN still work to install and uninstall the validate_password plugin as before, but generate warnings. The plugin form of validate_password is deprecated and will be removed in a future MySQL version. MySQL installations that use the plugin should transition to the component instead. See Transitioning to the Password Validation Component.

A usage difference between the component and plugin implementations of validate_password is that, whereas the plugin exposes system and status variables with names that begin with validate_password_, the component uses the convention of exposing variables with names that begin with the component name and a period. For example, the plugin exposes the validate_password_policy system variable and validate_password_dictionary_file_words_count status variable, whereas the corresponding variables exposed by the component are validate_password.policy and validate_password.dictionary_file_words_count. The plugin variables, like the plugin itself, are deprecated and will be removed in a future MySQL version.

Configuration Notes

• Replication: When calculating the binary log expiration period, any value that you specify for the deprecated expire_logs_days system variable is now ignored if binlog_expire_logs_seconds is set, and only the value of binlog_expire_logs_seconds is used. Previously, the effects of binlog_expire_logs_seconds and expire_logs_days were cumulative.

If you set a value for both system variables at startup, a warning message is issued stating that the value of expire_logs_days is ignored. You cannot set or change the value of one option dynamically while the other is set, and an error message is issued in this situation.

The default binary log expiration period of 30 days (as changed in MySQL 8.0.2) applies if neither binlog_expire_logs_seconds nor expire_logs_days has a value set at startup. If a value for either binlog_expire_logs_seconds or expire_logs_days is set at startup, this value overrides
the default binary log expiration period. To disable automatic purging of the binary log, you must set both `binlog_expire_logs_seconds` and `expire_logs_days` explicitly to 0 at startup. (Bug #26274274, Bug #86698)

- For RHEL, SLES, and Fedora RPMs, the default plugin directory for debug builds has been changed from `/usr/lib64/mysql/plugin` to `/usr/lib64/mysql/plugin/debug`. (Bug #27072155, Bug #88363)
- The new `WITH_LZMA` and `WITH_RE2` CMake options control whether to compile with the bundled or system LZMA and RE2 libraries.
- The `table_open_cache` system variable default variable has been increased from 2000 to 4000. This additionally changes the default for the autosized `table_definition_cache` system variable from 1400 to 2000.
- The `optimizer_trace_max_mem_size` system variable default value was changed from 16KB to 1MB to lessen the likelihood of optimizer trace truncation.
- The `log_error_verbosity` system variable default value was changed from 3 (error, warning, and information messages) to 2 (error and warning messages) to make `mysqld` error logging less verbose by default.

**Deprecation and Removal Notes**

- **InnoDB:** The `innodb_undo_tablespaces` configuration option is deprecated and will be removed in a future MySQL version.
- **Replication:** The `group_replication_primary_member` status variable has been deprecated and will be removed in a future MySQL version.
- **Replication:** The `group_replication_allow_local_disjoint_gtids_join` system variable has been removed.
- When the `libmysqld` embedded server library was removed in MySQL 8.0.1, the following `mysql_options()` options used only for `libmysqld` were not removed. They have now been removed.

```
MYSQL_OPT_GUESS_CONNECTION
MYSQL_OPT_USE_EMBEDDED_CONNECTION
MYSQL_OPT_USE_REMOTE_CONNECTION
MYSQL_SET_CLIENT_IP
```

(Bug #26712418)

- Generated columns no longer permit deprecated functions in the generation expression, to avoid problems when MySQL is upgraded to a version in which the deprecated functions have been removed. (Bug #26279884, Bug #86712)

**Function and Operator Notes**

- **Incompatible Change:** Previously, MySQL used the Henry Spencer regular expression library to support regular expression operators (REGEXP, RLIKE). Regular expression support has been reimplemented using International Components for Unicode (ICU), which provides full Unicode support and is multibyte safe. The `REGEXP_LIKE()` function performs regular expression matching in the manner of the `REGEXP` and `RLIKE` operators, which now are synonyms for that function. In addition, the `REGEXP_INSTR()`, `REGEXP_REPLACE()`, and `REGEXP_SUBSTR()` functions are available to find match positions and perform substring substitution and extraction, respectively. The `regexp_stack_limit`
and `regexp_time_limit` system variables provide control over resource consumption by the match engine.

MySQL source distributions bundle the ICU library, and the `WITH_ICU` CMake option controls whether to compile with the bundled or system ICU library. The `ICU_VERSION()` function returns the ICU library version.

For more information, see Regular Expressions. For information about ways in which applications that use regular expressions may be affected by the implementation change, see Regular Expression Compatibility Considerations.

- Two new SQL functions are available to provide digest information about SQL statements. Given an SQL statement as a string, `STATEMENT_DIGEST()` returns the statement digest hash value, and `STATEMENT_DIGEST_TEXT()` returns the normalized statement digest. See Encryption and Compression Functions.

**InnoDB Notes**

- The restriction enacted in MySQL 8.0.3 against renaming of columns in a parent foreign key has been lifted. (Bug #26659110, Bug #87490, Bug #25722221)

  References: See also: Bug #26334071.

**Installation Notes**

- For platforms that use systemd (see Managing MySQL Server with systemd), the data directory is initialized if empty at server startup. This might be a problem if the data directory is a remote mount that has temporarily disappeared: The mount point would appear to be an empty data directory, which then would be initialized as a new data directory. It is now possible to suppress this automatic initialization behavior. Specify the following line in the `/etc/sysconfig/mysql` file (create the file if it does not exist):

  ```
  NO_INIT=true
  ```

  (Bug #26595288, Bug #87287)

**Keyring Notes**

- MySQL now supports key migration between underlying keyring keystores. This enables DBAs to switch a MySQL installation from one keyring plugin to another. See Migrating Keys Between Keyring Keystores.

**Logging Notes**

- A new error log filter component, `log_filter_dragenet`, enables control over events written to the error log, based on user-defined rules specified as the value of the `dragenet.log_error_filter_rules` system variable. This new filtering capability is more flexible than is achievable using the built-in `log_filter_internal` filter component and `log_errorverbosity` system variable. For more information, see Types of Error Log Filtering.

- Messages written to the error log by the `log_sink_internal` log writer component now contain an error-ID indicator. This ID has a format of `[error_id]`. It follows the severity indicator and precedes the message text. For more information, see Error Log Message Format.

- When the `log_errorverbosity` system variable for error logging is set to filter out warning and information messages, selected important messages about non-error situations are now printed to error
logs as system messages. These messages were previously handled and labeled as errors so that they were logged when `log_error_verbosity=1` was set. The labeling function has now been decoupled from the handling function for these messages, so that the messages can be given an appropriate label.

System messages now have the label “System” in the MySQL error log. Other log writers that support custom labels might also use this convention. In logs that do not support custom labels, system messages are printed with the label used for the information level of severity. However, they are included even if the `log_error_verbosity` setting normally excludes messages at this level.

If your system has additional configuration to discard messages labeled as information level, or to redirect them to a different destination from messages labeled as errors, system messages do not override this behavior. If the log writer involved does not use the “System” label, system messages are now discarded or redirected along with other messages labeled as information level.

The messages that are now printed with the “System” label where supported include startup and shutdown messages, and some significant changes to settings, as follows:

- `ER_STARTING_AS`
- `ER_STARTUP`
- `ER_XA_STARTING_RECOVERY`
- `ER_XA_RECOVERY_DONE`
- `ER_NORMAL_SHUTDOWN`
- `ER_SHUTDOWN_COMPLETE`
- `ER_RPL_SLAVE_CONNECTED_TO_MASTER_REPLICATION_STARTED`
- `ER_RPL_SLAVE_CONNECTED_TO_MASTER_REPLICATION_RESUMED`
- `ER_SLAVE_CHANGE_MASTER_TO_EXECUTED`
- `ER_GTID_PURGED_WAS_CHANGED`
- `ER_GTID_EXECUTED_WAS_CHANGED`
- `ER_CHANGED_GTID_MODE`
- `ER_RPL_SLAVE_DUMP_THREAD_KILLED_BY_MASTER`

**Packaging Notes**

- All MySQL binary distributions now are linked against OpenSSL, including Community distributions, which previously were linked against yaSSL. In addition, OpenSSL is linked dynamically rather than statically, which enables substitution of alternative SSL libraries for use with MySQL if desired. For some platforms, binary distributions bundle OpenSSL libraries to ensure library availability:

  - Windows: Distributions bundle `libeay32.dll` and `ssleay32.dll`, which are installed in the same directory as MySQL binaries. Other libraries can be used by replacing those library files with alternatives, or by using some other library-selection method supported on Windows. (See [https://msdn.microsoft.com/en-us/library/7d83bc18.aspx](https://msdn.microsoft.com/en-us/library/7d83bc18.aspx).)

  - macOS: Distributions bundle `libssl.dylib` and `libcrypto.dylib`. MySQL binaries are linked to expect the libraries in the same directory, and symbolic links there point to the actual library locations.
MySQL 8.0 Release Notes

- Linux (for generic compressed tar file distributions only): Distributions bundle libssl.so and libcrypt.so, which are installed in the same directory as the libmysqlclient.so client library.

  (Bug #26272084, Bug #26134893, Bug #26927607, Bug #87996)

Performance Schema Notes

- The Performance Schema now uses SHA-256 hashes for statement digests rather than MD5 hashes. To accommodate the increased storage required for SHA-256 values, DIGEST columns in Performance Schema tables are widened from VARCHAR(32) to VARCHAR(64). (Bug #26727443)

- The Performance Schema setup_timers table has been removed, as has the TICK row in the performance_timers table. (Bug #18296337)

Security Notes

- Incompatible Change: Passwords are now restricted to a maximum of 256 characters for the caching_sha2_password and sha256_password authentication plugins, and for the PASSWORD() function when old_passwords=2. Also, the number of password hashing rounds is capped to limit CPU time used. (Bug #27099029, Bug #27194270)

- Incompatible Change: The caching_sha2_password and sha256_password authentication plugins provide more secure password encryption than the mysql_native_password plugin, and caching_sha2_password provides better performance than sha256_password. Due to these superior security and performance characteristics of caching_sha2_password, it is now the preferred authentication plugin, and is also the default authentication plugin rather than mysql_native_password. This change affects both the server and the libmysqlclient client library:

  - For the server, the default value of the default_authentication_plugin system variable changes from mysql_native_password to caching_sha2_password.

  - The libmysqlclient library treats caching_sha2_password as the default authentication plugin rather than mysql_native_password.

This change has the following implications:

- The change affects only the authentication plugin used for creating new MySQL accounts. For accounts already existing in an upgraded installation, their authentication plugin remains unchanged.

- Clients that use an account that authenticates with caching_sha2_password must use either a secure connection (made using TCP using TLS/SSL credentials, a Unix socket file, or shared memory), or an unencrypted connection that supports password exchange using an RSA key pair. This security requirement does not apply to mysql_native_password, so the switch to caching_sha2_password may require additional configuration (see Caching SHA-2 Pluggable Authentication). However, client connections in MySQL 8.0 prefer use of TLS/SSL by default, so clients that already conform to that preference may need no additional configuration.

- Because caching_sha2_password is also now the default authentication plugin in the libmysqlclient client library, authentication requires an extra round trip in the client/server protocol for connections from MySQL 8.0 clients to accounts that use mysql_native_password (the previous default authentication plugin), unless the client program is invoked with a --default-auth=mysql_native_password option.
• Incompatibility: Clients and connectors that have not been updated to know about `caching_sha2_password` cannot connect to accounts that authenticate with `caching_sha2_password` because they do not recognize this plugin as valid. To work around this issue, relink clients against `libmysqlclient` from MySQL 8.0.4 or higher, or obtain an updated connector that recognizes `caching_sha2_password`.

• Incompatibility: Clients and connectors that have not been updated to know about `caching_sha2_password` may have trouble connecting to a MySQL 8.0 server configured with `caching_sha2_password` as the default authentication plugin, even to use accounts that do not authenticate with `caching_sha2_password`. This issue occurs because the server specifies the name of its default authentication plugin to clients. If a client or connector is based on a client/server protocol implementation that does not gracefully handle an unrecognized default authentication plugin, it may fail with an error.

For more information about the more prominent role of `caching_sha2_password`, including discussion of potential compatibility issues and solutions, see [caching_sha2_password as the Preferred Authentication Plugin](https://dev.mysql.com/doc/refman/8.0/en/caching_sha2_password.html).

Additionally, replication slaves are now able to connect to masters using RSA key pair-based password exchange. This RSA capability applies for accounts that authenticate with the `caching_sha2_password` or `sha256_password` authentication plugin. (Previously, slaves could authenticate with those plugins, but required a secure connection and could not use RSA password exchange.) The following changes enable use of this new slave RSA capability:

• The `CHANGE MASTER TO` statement has clauses for specifying RSA public key information.

• For Group Replication, the `group_replication_recovery_public_key_path` and `group_replication_recovery_get_public_key` system variables serve the same purpose.

• The `SHOW SLAVE STATUS` statement and the Performance Schema `replication_connection_configuration` table display replication slave RSA public key information.

More programs now support a `--server-public-key-path` option to enable specifying a client-side file containing the public key for RSA key pair-based password exchange, for accounts that authenticate with the `caching_sha2_password` or `sha256_password` authentication plugin: `mysqladmin`, `mysqlbinlog`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlpump`, `mysqldump`, `mysql_upgrade`. Previously, only `mysql` and `mysqltest` supported `--server-public-key-path`. The `--server-public-key-path` option is an alternative to `--get-server-public-key`, and may be used when a client-side file is available that contains a copy of the RSA public key required by the server, rather than sending a request for the public key to the server.

The new `mysql_reset_server_public_key()` C API function clears any cached copy of the server RSA public key from the client library. See `mysql_reset_server_public_key()`.


The new `caching_sha2_password_rsa_public_key` status variable exposes the public key used by the `caching_sha2_password` authentication plugin for RSA key pair-based password exchange. (Bug #26751594)
• The linked OpenSSL library for MySQL Server has been updated to version 1.0.2n. Issues fixed in
the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html. (Bug
#27212666, Bug #27236394)

• Certificates automatically generated by `mysqld` and `mysql_ssl_rsa_setup` now use X509 v3 rather
than v1. (Bug #26521654)

Server Administration

• A new `RESTART` SQL statement is available that enables restarting a MySQL server instance from within
a client session. This feature is available for platforms on which a monitoring process is able to detect
a server shutdown performed for restart purposes: Windows (`mysqld` started as a Windows service or
standalone), and Unix and Unix-like systems that use systemd or `mysqld_safe` to manage `mysqld`. See `RESTART Syntax`.

Spatial Data Support

• **Incompatible Change:** These spatial import functions now return an error for data in a
geospatial data reference system for which the coordinates are outside the legal range of
values: `ST_GeomCollFromText()`, `ST_GeomCollFromTxt()`, `ST_GeometryCollectionFromText()`,
`ST_GeometryCollectionFromWKB()`, `ST_GeometryFromText()`, `ST_GeometryFromWKB()`, `ST_GeomFromGeoJSON()`,
`ST_GeomFromText()`, `ST_GeomFromWKB()`, `ST_LineFromText()`, `ST_LineFromWKB()`,
`ST_LineStringFromText()`, `ST_LineStringFromWKB()`, `ST_MLineFromText()`,
`ST_MLineFromWKB()`, `ST_MPointFromText()`, `ST_MPointFromWKB()`, `ST_MPolyFromText()`,
`ST_MPolyFromWKB()`, `ST_MultiLineStringFromText()`, `ST_MultiLineStringFromWKB()`,
`ST_MultiPointFromText()`, `ST_MultiPointFromWKB()`, `ST_MultiPolygonFromText()`,
`ST_MultiPolygonFromWKB()`, `ST_PointFromText()`, `ST_PointFromWKB()`,
`ST_PolyFromText()`, `ST_PolyFromWKB()`, `ST_PolygonFromText()`, `ST_PolygonFromWKB()`. (`ST_LatFromGeoHash()`, `ST_LongFromGeoHash()`, and `ST_PointFromGeoHash()` do not have
this behavior because they cannot represent out-of-range values.)

`ST_GeomFromGeoJSON()` also now returns an error if a `crs` member that specifies an SRID different
from the top-level object SRID is found at a lower level of the GeoJSON document.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
</table>
| If spatial data contains SRID values that refer to a geographic spatial reference
system and the data coordinates are out of range, existing queries using these
functions will return different results, compared to previous MySQL versions. |

(Bug #26941370, Bug #88031)

• **Incompatible Change:** Previously, these spatial functions ignored the spatial reference system (SRS)
for geometry arguments and computed results on a Cartesian plane. They now support computations
for geometry arguments that specify a geographic SRS: `ST_Distance_Sphere()`, `ST_IsSimple()`,
`ST_IsValid()`, `ST_Length()`.

Previously, these spatial functions ignored the SRS for any geometry arguments and computed
results on a Cartesian plane. They now produce an error when invoked with geometry
arguments that specify a geographic SRS: `ST_Area()`, `ST_Buffer()`, `ST_Centroid()`,
`ST_ConvexHull()`, `ST_Difference()`, `ST_Envelope()`, `ST_Intersection()`, `ST_IsClosed()`,
`ST_MakeEnvelope()`, `ST_Simplify()`, `ST_SymDifference()`, `ST_Union()`, `ST_Validate()`.

Previously, these spatial functions permitted geometry arguments with an undefined
SRS. They now produce an error when invoked with geometry arguments that have
an undefined SRS: `ST_Dimension()`, `ST_Distance_Sphere()`, `ST_EndPoint()`, `ST_ExteriorRing()`, `ST_GeometryN()`, `ST_GeometryType()`, `ST_InteriorRingN()`, `ST_IsEmpty()`, `ST_IsSimple()`, `ST_IsValid()`, `ST_Length()`, `ST_NumGeometries()`, `ST_NumInteriorRing()`, `ST_NumInteriorRings()`, `ST_NumPoints()`, `ST_PointN()`, `ST_StartPoint()`, `ST_SwapXY()`, `ST_X()`, `ST_Y()`.

Previously, the `ST_GeoHash()` spatial function accepted points with any SRID. `ST_GeoHash()` now accepts only points with SRID 0 or 4326.

**Note**
If spatial data contains geometry values that now are interpreted differently by the functions just listed, existing queries using these functions will return different results, compared to previous MySQL versions.

- **Incompatible Change:** For the following spatial functions that take an SRID argument, they now return an error if the SRID does not exist: `ST_GeomCollFromText()`, `ST_GeomCollFromTxt()`, `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromText()`, `ST_GeometryCollectionFromWKB()`, `ST_GeometryFromGeoJSON()`, `ST_GeomFromText()`, `ST_GeomFromWKB()`, `ST_LineFromText()`, `ST_LineFromWKB()`, `ST_LineStringFromText()`, `ST_LineStringFromWKB()`, `ST_MLineFromText()`, `ST_MLineFromWKB()`, `ST_MPPointFromText()`, `ST_MPPointFromWKB()`, `ST_MPolyFromText()`, `ST_MPolyFromWKB()`, `ST_MultiLineStringFromText()`, `ST_MultiLineStringFromWKB()`, `ST_MultiPointFromText()`, `ST_MultiPointFromWKB()`, `ST_MultiPolygonFromText()`, `ST_MultiPolygonFromWKB()`, `ST_PointFromGeoHash()`, `ST_PointFromText()`, `ST_PointFromWKB()`, `ST_PolyFromText()`, `ST_PolyFromWKB()`, `ST_PolygonFromText()`, `ST_PolygonFromWKB()`. (`ST_SRID()` already had this behavior.)

**Note**
If spatial data contains SRID values that refer to a nonexistent spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

- The server now prints a warning if a **SPATIAL** index is created that the optimizer will never use. (Bug #27015964)
- The EPSG Dataset containing spatial reference system data for spatial calculations has been upgraded from version 9.0 to 9.2. (Bug #26711258, Bug #87564, Bug #26990896)
- MySQL now supports `CREATE SPATIAL REFERENCE SYSTEM` and `DROP SPATIAL REFERENCE SYSTEM` statements for creating and removing spatial reference system (SRS) definitions in the data dictionary. See `CREATE SPATIAL REFERENCE SYSTEM Syntax`, and `DROP SPATIAL REFERENCE SYSTEM Syntax`. With the implementation of these statements, direct access to the `mysql.st_spatial_reference_systems` data dictionary table using statements such as `SELECT`, `INSERT`, and `UPDATE` is not needed, and is no longer permitted. Read access to SRS definitions remains available through the `INFORMATION_SCHEMA ST_SPATIAL_REFERENCE_SYSTEMS` table, as before.

**Test Suite Notes**
- Documentation for the MySQL Test Suite is now maintained in the MySQL source tree using Doxygen (see the MySQL Server Doxygen documentation, available at [https://dev.mysql.com/doc/index-other.html](https://dev.mysql.com/doc/index-other.html)). The related Unix man pages that previously were produced from the old test suite manual are no longer updated and have gone out of date. Consequently, they are no longer included in MySQL distributions. (Bug #27021754)
MySQL 8.0 Release Notes

- The MySQL test suite now includes CRC32() tests. Thanks to Daniel Black for the patch. (Bug #26495791, Bug #87136)
- mysqltest now accepts an optional retry argument for these commands: `copy_file`, `copy_files_wildcard`, `file_exists`, `move_file`, `remove_file`, and `remove_files_wildcard`. Usage is as follows:

  ```
  --file_operation file_name ... [retry]
  ```

  When the retry argument is specified, the command retries a failed operation the given number of times at one-second intervals. (Bug #24671890)

**X Plugin Notes**

- X Plugin could incorrectly remove quotes from parameters that were passed to functions expecting JSON strings, causing an error. (Bug #26906519)
- After an index was created on a collection, X Plugin would list the collection as a relational table, and no longer list it as a collection. (Bug #26906487)
- X Protocol displayed the DATE type in the same way as the DATETIME type. The two data types are now treated differently. (Bug #26647488)
- mysqlxtest did not display fractional seconds in DATETIME values correctly. (Bug #26638422)
- X Plugin returned a column metadata field showing fractional digits for a JSON column, for which that information was irrelevant. The field is no longer returned. (Bug #26258481)
- X Plugin handled user account lookup incorrectly if the IP address resolved to an empty host name. (Bug #26042786)
- X Plugin can now be configured to abort any authenticated connections which are considered idle. The following system variables have been added:
  - `mysqlx_interactive_timeout`
  - `mysqlx_read_timeout`
  - `mysqlx_wait_timeout`
  - `mysqlx_write_timeout`
- X Plugin now supports Caching SHA-2 Pluggable Authentication. To enable the support, issue:

  ```
  INSTALL PLUGIN mysqlx_cache_cleaner SONAME 'mysqlx.so';
  ```

  This plugin maintains the cache and removes all entries for accounts that were modified.

  References: See also: Bug #27213213, Bug #27042109.

**Functionality Added or Changed**

- **InnoDB**: Obsoleted InnoDB system tables that are no longer required after upgrading to MySQL 8.0.3 or later are now dropped after a successful upgrade. (Bug #26757171)
- **InnoDB**: Support was added to automatically upgrade data dictionary table definitions when starting the MySQL server after upgrading MySQL binaries to a new version. At startup, the data dictionary version of the server is compared to the version information stored in the data dictionary to determine if data
dictionary tables should be upgraded. If an upgrade is necessary and supported, the server creates
data dictionary tables with updated definitions, copies persisted metadata to the new tables, atomically
replaces the old tables with the new ones, and reinitializes the data dictionary. If an upgrade is not
necessary, startup continues without updating the data dictionary tables.

The `mysqld --no-dd-upgrade` option can be used to prevent automatic upgrade of data dictionary
tables at startup.

- **InnoDB**: Moving or restoring tablespace files to a new location while the server is offline is supported by
  the new `--innodb-directories` option, which defines directories to scan at startup for tablespace
  files. For more information, see Moving Tablespace Files While the Server is Offline.

With the introduction of the `--innodb-directories` feature, the location of file-per-table and general
tablespace files created with an absolute path or in a location outside of the data directory should be
added to the `innodb_directories` argument value. Otherwise, InnoDB is not able to locate these
files during recovery. To view tablespace file locations, query the `INFORMATION_SCHEMA.FILES` table:

```
mysql> SELECT TABLESPACE_NAME, FILE_NAME FROM INFORMATION_SCHEMA.FILES \G
```

The `CREATE TABLESPACE ... ADD DATAFILE` statement now requires that a specified directory
is known to InnoDB. Known directories include those implicitly and explicitly defined by the
`innodb_directories` option.

`innodb_directories` replaces `innodb_scan_directories`, which has been removed.

- **InnoDB**: InnoDB now supports partial update of large object (LOB) data stored in external fields
  outside of clustered index pages, including JSON documents updated using `JSON_SET()` and
  `JSON_REPLACE()`. Previously, LOB values could only be read or modified in full, and updates of JSON
column values were done by completely removing the previous document and writing the new one in its
place.

Partial update uses an internal LOB index, or ZLOB index in the case of compressed LOB data, which is
created and stored in one or more LOB pages when a row containing LOB data is inserted or updated.
Thus, partial fetch and update of LOB data is supported for newly inserted rows, but existing rows
containing LOB data must be updated to add partial fetch and update support. The addition of LOB index
data increases the storage space required by LOB values by a small percentage.

This feature adds the following InnoDB page types for storing uncompressed and compressed LOB
data:

- `FIL_PAGE_TYPE_LOB_INDEX`
- `FIL_PAGE_TYPE_LOB_DATA`
- `FIL_PAGE_TYPE_LOB_FIRST`
- `FIL_PAGE_TYPE_ZLOB_FIRST`
- `FIL_PAGE_TYPE_ZLOB_DATA`
- `FIL_PAGE_TYPE_ZLOB_INDEX`
- `FIL_PAGE_TYPE_ZLOB_FRAG`
- `FIL_PAGE_TYPE_ZLOB_FRAG_ENTRY`
MySQL 8.0 Release Notes

- **InnoDB:** The storage engine private field of the `mysql.tablespaces` data dictionary table and page 0 of `InnoDB` tablespace files now store MySQL server version and tablespace version information. The MySQL server version is the version that created the tablespace, the version into which the tablespace was imported, or the version of the last major MySQL upgrade. The tablespace version tracks tablespace format changes. `SERVER_VERSION` and `SPACE_VERSION` fields were added to the `INFORMATION_SCHEMA.INNODB_TABLESPACES` table to provide access to this data.

- **Replication:** Host names can now be specified as part of a whitelist for group replication connections, using the `group_replication_ip_whitelist` system variable. Host names support CIDR notation. Host names that resolve to IPv6 addresses are not supported.

For host names, name resolution takes place only when a connection request is made by another server. A host name that cannot be resolved is not considered for whitelist validation, and a warning message is written to the error log. Forward-confirmed reverse DNS (FCrDNS) verification is carried out for resolved host names.

**Warning**

Host names are inherently less secure than IP addresses in a whitelist. FCrDNS verification provides a good level of protection, but can be compromised by certain types of attack. Specify host names in your whitelist only when strictly necessary, and ensure that all components used for name resolution, such as DNS servers, are maintained under your control. You can also implement name resolution locally using the hosts file, to avoid the use of external components.

- **JSON:** Added the `JSON_TABLE()` function, which accepts JSON data and returns it as a relational table whose columns are as specified. This virtual table can be accessed using standard SQL mechanisms. This function has the syntax `JSON_TABLE(expr, path COLUMNS column_list) [AS] alias`, where `expr` is an expression that returns JSON data, `path` is a JSON path applied to the source, and `column_list` is a list of column definitions. Columns can be of the types `FOR ORDINAL`, `PATH`, `EXISTS PATH`, and `NESTED PATH` or `NESTED`, as described in the following list:

  - **FOR ORDINAL:** The column is a counter, similar to an `AUTO_INCREMENT` column.
  - **PATH:** The column holds a scalar value using the specified JSON path. `ON ERROR` and `ON EMPTY` options are supported for handling illegal values (such as non-scalars) and empty values, respectively.
  - **EXISTS PATH:** The column value is 1 if a match exists for the specified JSON path, and 0 otherwise.
  - **NESTED PATH:** Nested objects or arrays in JSON data found in the given JSON path are flattened into a single row along with the JSON values from the parent object or array. The `PATH` keyword is optional.

Two simple examples are shown here:

```sql
mysql> SELECT * FROM JSON_TABLE( '[1, 3, 5, 7, 9]', '$[*]' COLUMNS(rowid FOR ORDINALITY, col VARCHAR(50) PATH '$') ) AS t1;
+-------+------+
| rowid | col  |
+-------+------+
|     1 | 1    |
|     2 | 3    |
|     3 | 5    |
```
The JSON source expression can be any expression that yields a valid JSON document, including a JSON literal, a table column, or a function call that returns JSON such as JSON_EXTRACT(t1, data, '$.post.comments'). For complete syntax and other information about this function, see JSON Table Functions.

- The -DWITH_ASAN_SCOPE CMake option enables the AddressSanitizer -fsanitize-address-use-after-scope Clang flag for use-after-scope detection. The default is off. To use this option, -DWITH_ASAN must also be enabled. (Bug #27095089)

- Handling of empty JSON documents has been made more robust. (Bug #26787468, Bug #87734)

- MySQL 8.0.3 imposed a restriction against ALTER TABLE RENAME on tables in a foreign key relationship if a LOCK TABLES was active. This restriction has been lifted. (Bug #26647340, Bug #87467)

- If the server PID file is configured to be created in a world-writable location, the server now issues a warning suggesting use of a more secure location. (Bug #26585560)

- For constructs using _charset character set introducers (for example, _utf8mb4 'abc'), lookup performance was improved for mapping the introducer onto the proper character set. (Bug #25680866, Bug #85331)

### Bugs Fixed

- **Important Change:** The following changes are made to the PERIOD_ADD() and PERIOD_DIFF() functions:
  
  - A period value used with one of these functions may not be negative.
  
  - The month part of a period value may not be equal to 0.

A period value used with one of these functions for which at least one of these conditions is true now causes the function to fail with an error. (Bug #27004699, Bug #27004729)
• **Important Change:** The `LEAST()` and `GREATEST()` functions no longer attempt to infer a context for their arguments from expressions in which they are used. For example, `LEAST('11', '45', '2')` returns '11', but `LEAST('11', '45', '2') + 0` treated the function arguments as integers rather than as strings, and returned 2. Now these functions always evaluate their arguments strictly according to type, and any data type coercion due to their inclusion in an expression is performed only on the result returned by the function. This means that the expression `LEAST('11', '45', '2') + 0` now evaluates to '11' + 0, and thus to the integer value 11.

This change has been made due to the following considerations:

• Rules for deriving the context were not always clear or consistent.

• The results of these functions when used in expressions were not consistent with the results of `COALESCE()`, or of a `UNION` query.

Applications that use these functions within expressions should be checked to make sure that they do not depend on the previous behavior, and updated if they do so. (Bug #83895, Bug #25123839)

• **InnoDB:** An `ALTER TABLE` operation that added a foreign key constraint referencing a table with generated virtual columns raised an assertion. (Bug #27189701)

• **InnoDB:** Concurrent XA transactions that ran successfully to the XA prepare stage on the master conflicted when replayed on the slave, resulting in a lock wait timeout in the applier thread. The conflict was due to the GAP lock range which differed when the transactions were replayed serially on the slave. To prevent this type of conflict, GAP locks taken by XA transactions in `READ COMMITTED` isolation level are now released (and no longer inherited) when XA transactions reach the prepare stage. (Bug #27189701, Bug #25866046)

• **InnoDB:** A `DROP DATABASE` operation raised an assertion due to a missing general tablespace data file. (Bug #27151163)

• **InnoDB:** On Windows, an operation that altered a table partition raised an assertion. The table name was not parsed correctly. (Bug #27075816)

• **InnoDB:** A `TRUNCATE TABLE` operation on a temporary table raised an assertion. (Bug #27073280)

• **InnoDB:** A call to a recovery-related function during the post-DDL phase of a DDL operation raised an assertion. (Bug #27041487, Bug #88263)

• **InnoDB:** Adding a spatial index that referenced a column with an SRID attribute returned an error. (Bug #27021029)

• **InnoDB:** A table with a 64-character foreign key name caused an upgrade failure. Foreign key names up to 64 characters in length should be permitted. (Bug #27014308, Bug #88196)

• **InnoDB:** The *InnoDB* recovery process failed with a tablespace size error for a compressed table that was upgraded from MySQL 5.7 to MySQL 8.0.

  The tablespace file for a compressed table is now created using the physical page size instead of the *InnoDB* page size, which makes the initial size of a tablespace file for an empty compressed table smaller than in previous MySQL releases. (Bug #27014083, Bug #88195)

• **InnoDB:** An orphan .frm file caused an upgrade failure, and subsequent upgrade attempts were unsuccessful due to a full-text search auxiliary table that was renamed during the first upgrade attempt. (Bug #26995951)

• **InnoDB:** Unnecessary tablespace fetch and cache update operations caused a server startup delay. (Bug #26995951)
References: This issue is a regression of: Bug #26832347.

- **InnoDB**: Workarounds introduced to address conflicting serialized dictionary information (SDI) inserts during concurrent CREATE TABLE operations were removed. (Bug #26995534)
  References: See also: Bug #26539665.

- **InnoDB**: A “no space left on device” error reported an invalid error message. (Bug #26960345)

- **InnoDB**: During a fast shutdown, InnoDB attempted to write dynamic metadata to the data dictionary after files were closed, resulting in an initialization failure due pending I/O on the data dictionary tablespace. (Bug #26950659)

- **InnoDB**: A stack overflow error was encountered on startup after upgrading to MySQL 8.0.4 due to repeated attempts to load an evicted InnoDB system table. (Bug #26945437, Bug #88042)

- **InnoDB**: Importing a compressed table raised an assertion. The operation used the clustered index of the table instead of the serialized dictionary information (SDI) index to transform SDI pages. (Bug #26938297)

- **InnoDB**: In debug builds, failed temporary table creation during a REPLACE operation raised an invalid assertion. (Bug #26919378, Bug #26958868)

- **InnoDB**: DROP DATABASE failed if database tables were created in a general tablespace. General tablespace flags were registered incorrectly causing the serialized dictionary information (SDI) operation to fail. (Bug #26834496)

- **InnoDB**: With binary logging enabled, an ALTER TABLESPACE ... RENAME operation failed with a “cannot find space” error. (Bug #26832347)

- **InnoDB**: An operation that failed to add an index raised an invalid adaptive hash index assertion. (Bug #2688968)

- **InnoDB**: A valid table row type value read from the data dictionary raised an invalid assertion. (Bug #26773152)

- **InnoDB**: Starting an upgrade with innodb_force_recovery=5 initialized InnoDB background threads but did not exit the threads gracefully when an error was encountered. Upgrading with a nonzero innodb_force_recovery setting is no longer permitted. (Bug #26766632)

- **InnoDB**: A failed CREATE TEMPORARY TABLE statement left an entry in INFORMATION_SCHEMA.INNODB_TEMP_TABLE_INFO. The in-memory table object was not freed. (Bug #26765438)

- **InnoDB**: InnoDB looked up the name of a virtual column in the wrong dict_table_t array when attempting to locate a qualifying index for a foreign key. (Bug #26764604)

- **InnoDB**: Attachable read-write transactions that update the table_stats and index_stats data dictionary tables attempted to update the same row, causing a deadlock. (Bug #26762517)

- **InnoDB**: During recovery, the tablespace name in an in-memory tablespace object was defined using the file name character set instead of table name character set, resulting in a “missing tablespace” error. (Bug #26761960)

- **InnoDB**: Bootstrap code did not reserve the first 1024 table IDs for data dictionary tables. (Bug #26757227)
• **InnoDB:** Multiple updates from different clients on a partitioned table caused an unexpected lock wait timeout due to an incorrectly set lock type. (Bug #26731025, Bug #87619)

• **InnoDB:** An asynchronous rollback thread that attempted to acquire a metadata lock was interrupted, but the resulting error was not returned to the server. This issue was addressed by removing the metadata lock acquisition, which was not necessary for asynchronous rollback.

Only in-memory tables were checked when opening tables for undo processing. The data dictionary is now checked as well, in case tables are not present in memory. (Bug #26678883)

• **InnoDB:** An assertion was raised when attempting to open a full-text auxiliary table with a name that was longer than expected. (Bug #26649020)

• **InnoDB:** Data dictionary table open functions did not properly handle table and schema name character set conversion, resulting in an error during recovery. (Bug #26640776)

• **InnoDB:** A transaction end_stmt() function was not called in some ALTER TABLE ... PARTITION scenarios, resulting in a timeout. (Bug #26629790, Bug #25886814)

• **InnoDB:** Acquiring a metadata lock on the serialized diction information (SDI) table during the commit phase of a DDL operation would fail due to a lock wait timeout or halting of the query. (Bug #26628126)

• **InnoDB:** Redo logs for dynamic metadata updates were not considered when checking redo log margin. Also, in read-only mode, the innodb_dynamic_metadata data dictionary table was opened unnecessarily for writing of metadata from the redo log. (Bug #26589535)

• **InnoDB:** An unexpected error occurred after a failed attempt to install the memcached plugin. (Bug #26588738)

• **InnoDB:** The state of a buffer pool page was altered by another thread while a buffer pool resize operation was in progress. (Bug #26588537)

• **InnoDB:** Debug functions that assert for conflicting locks did not account for transaction locks that are to be committed or rolled back. (Bug #26562371)

• **InnoDB:** Variance-Aware Transaction Scheduling (VATS) functionality that updates the age of waiting record locks failed to ignore table locks, causing an assertion failure. (Bug #26538702)

• **InnoDB:** A DDL operation that created or modified a table partition unintentionally altered the row format of other partitions, resulting in a row format mismatch. (Bug #26535746)

• **InnoDB:** An ALTER TABLE operation caused the server to halt. (Bug #26492721)

• **InnoDB:** The innodb_table_stats data dictionary table was not updated with new partition names when renaming a partitioned table. (Bug #26390658, Bug #86927)

• **InnoDB:** Due to a regression introduced in MySQL 8.0.0, the innodb_change_buffering configuration option could not be set dynamically. (Bug #26389442)

• **InnoDB:** The online log for a freed index was accessed while rolling back a concurrent UPDATE statement during an online DDL operation. (Bug #26334475)

• **InnoDB:** A REPLACE operation on a table with a secondary index on the prefix of a virtual column raised an assertion. (Bug #26330279)

• **InnoDB:** Setting tmpdir to the root of a drive caused “Invalid (old?) table or database name” error messages to be printed to the error log. (Bug #26299984, Bug #86737)
• **InnoDB:** A race condition occurred during an INFORMATION_SCHEMA query when attempting to check the transaction state without acquiring a transaction mutex. (Bug #26299705)

• **InnoDB:** A FLUSH TABLES operation failed to drop an aborted index. While removing the table from the cache, the clustered index was dropped prior to checking for the aborted index. (Bug #26256456, Bug #86607)

• **InnoDB:** For InnoDB tables, CREATE TABLE ... LIKE did not respect the innodb_file_per_table system variable setting, and SHOW CREATE TABLE displayed a TABLESPACE clause even though the user specified no explicit tablespace during table creation. (Bug #26199233, Bug #86589)

• **InnoDB:** An iterative approach to processing foreign cascade operations resulted in excessive memory use. (Bug #26191879, Bug #86573)

  References: This issue is a regression of: Bug #16244691.

• **InnoDB:** The lock acquisition sequence for a buffer pool eviction operation that evicts compressed pages was incorrect. (Bug #25972975)

• **InnoDB:** Metadata locks were released while data dictionary objects were still in use. (Bug #25928984)

• **InnoDB:** innochecksum returned a Valgrind error when run on InnoDB files with a 1K compressed page size. (Bug #25922124, Bug #85993)

• **InnoDB:** A kill thread failed to close the socket of another thread that was executing a TRUNCATE TABLE operation, causing an assertion. (Bug #25887335, Bug #85925)

• **InnoDB:** An INSERT operation on a table with a spatial index raised an assertion due to a failure that occurred during a lock conflict check. (Bug #25729649)

• **InnoDB:** A debug sync point intended for user tables was activated for data dictionary tables. (Bug #25508568)

• **InnoDB:** A server-side check was added to prevent a foreign key constraint from being placed on the base column of a generated stored column. (Bug #25339192)

• **InnoDB:** Warnings that should only appear in debug builds of MySQL were printed to the error log when the length of the history list exceeded 2000000. (Bug #24296076, Bug #82213)

• **InnoDB:** Attempting to reduce the buffer pool size to less than the buffer pool chunk size did not report a warning. (Bug #23590280)

• **InnoDB:** A “wrong key column” error was added to address an unsupported index creation scenario. (Bug #22486025)

• **InnoDB:** Full-text search on indexed columns that use a binary collation did not return case-sensitive matches. (Bug #21625016, Bug #78048)

• **Packaging:** When trying to install MySQL Server on Fedora 27 using the MySQL Yum repository, installation failed due to a conflict with the native mariadb-connector-c-devel package. With this fix, the appropriate “obsoletes” have been added for that and other native packages. (Bug #26963839)

• **Partitioning:** When creating a partitioned table using an implicit tablespace, the effect is to place each partition in its own tablespace, with no designated tablespace for the table as a whole. Since serialized dictionary information (SDI) was stored in all tablespaces used by a given table, the cost of storing it in a table with many tablespaces became prohibitive. This problem is solved by including only the tablespace for the first partition in the set of tablespaces used to store the SDI. (Bug #26762973)
Partitioning: An assertion could be raised on `CREATE VIEW` on partitioned tables when the server tried to prune partitions of the underlying tables. (Bug #26659699)

Partitioning: When renaming a partitioned table, the table statistics were not updated with the new partition names. (Bug #86074, Bug #25953183)

Replication; JSON: For row-based replication, partial updates to JSON documents were not applied if the server variable `binlog_row_value_options=PARTIAL_JSON` (introduced in MySQL 8.0.3) was not specified on the replication slave, as well as on the master. Now, a replication slave applies partial updates to JSON documents whenever these are received from the master, whether or not the slave has `binlog_row_value_options=PARTIAL_JSON` in its own settings. (Bug #26762675)

Replication: The function `set_unknown_error()` in the `Binlog_sender` class has been renamed to `set_unknown_error()`. Thanks to Simon Mudd for the fix (and also for the typo fix in Bug 88149). (Bug #27149075, Bug #88559)

References: See also: Bug #26996065, Bug #88149.

Replication: When you invoke `mysqld` with the `--initialize` or `--initialize-insecure` option to initialize the data directory, a warning message is no longer issues regarding the availability of the `mysql.gtid_executed` table, which should not be available at that stage. Also, the message formerly issued as a warning regarding the generation of a new UUID is now issued as a note, because the generation of a new UUID is normal in that situation. (Bug #27115183)

Replication: In MySQL 8.0.3, the default base name for the binary log files and index file was `host_name-bin`, using the name of the host machine. This default name was used if the `--log-bin` option was not supplied, and also if the `--log-bin` option was supplied with no string or with an empty string. From MySQL 8.0.4, if you do not supply the `--log-bin` option, MySQL now uses `binlog` as the default base name for the binary log files and index file. In releases before MySQL 8.0.3, there was no binary log with that configuration, so there is no incompatibility with existing binary logs at upgrade. However, for compatibility with existing binary logs from releases before MySQL 8.0.3, if you supply the `--log-bin` option with no string or with an empty string, the base name defaults to `host_name-bin`, using the name of the host machine.

The warning messages that were previously issued at startup if you did not specify a binary log file name using the `--log-bin` option (ER_LOG_BIN_BETTER_WITH_NAME) and if you did not specify a server ID using the `--server-id` option (ER_WARN_NO_SERVERID_SPECIFIED) are now issued as informational messages. A warning message is still issued if replication is attempted with a nonunique server ID. (Bug #27082922)

Replication: In the `Gtid_log_event` that precedes every GTID transaction in the binary log file, the `transaction_length` field used 8 bytes for transactions with 16777216 bytes or more, when it should have used the maximum permitted 9 bytes. (Bug #26993433)

Replication: The fix for Bug #22671846 was missing from MySQL version 8.0.3. (Bug #26985976)

Replication: The fix for Bug #26117735 (MySQL Bug #86288) could cause a debug assertion when running `mysqlbinlog` with the `--read-from-remote-server` option and the `--rewrite-db` option, depending on the database names specified in the rewrite rule. The issue has now been corrected. (Bug #26878022)

Replication: With MySQL compiled using yaSSL, and semisynchronous replication in use, a deadlock could be caused by incorrect handling of acknowledgement packets. Multiple acknowledgement packets can be read together by yaSSL, but the receiver thread for semisynchronous replication only handled the
first acknowledgement packet seen after polling. Now, the receiver thread handles all acknowledgement packets that are present in the buffer. (Bug #26865538)

- **Replication:** With semisynchronous replication in use, if `RESET MASTER` was issued while an active transaction was waiting for an acknowledgement from the slave, the count of waiting sessions in the `Rpl_semi_sync_master_wait_sessions` server status variable was incorrect after the wait was completed. (Bug #26748533)

- **Replication:** The `--log-slave-updates` and `--slave-preserve-commit-order` options require binary logging. If you specify these options and also disable binary logging using the `--skip-log-bin` or `--disable-log-bin` option, a warning or error message is issued. The `--skip-log-bin` and `--disable-log-bin` options now disable the `--log-slave-updates` and `--slave-preserve-commit-order` options by default, so when those options are not specified, the warning or error message is not issued. (Bug #26666259)

- **Replication:** `XA ROLLBACK` statements that failed because an incorrect transaction ID was given, could be recorded in the binary log with the correct transaction ID, and could therefore be actioned by replication slaves. A check is now made for the error situation before binary logging takes place, and failed `XA ROLLBACK` statements are not logged. (Bug #26618925, Bug #87393)

- **Replication:** The thread where the Group Replication plugin was started was not being correctly killed. This made it impossible to stop or start the plugin after killing the thread where Group Replication was started. (Bug #26435775)

- **Replication:** For the NDB storage engine, when the slave used hashing for searches of rows (which is included by default in the setting for the `slave_rows_search_algorithms` system variable from MySQL 8.0.2), the table used to store the row hashes was not cleaned up correctly after records were removed on the slave. The issue was caused by a variant error value returned by the NDB storage engine, which has now been corrected to the expected value. (Bug #26434966)

- **Replication:** MySQL internal administration commands that update replication-specific repository tables, for example during a replication synchronization check using the `mysqlrplsync` utility, can now bypass read locks. This enables such commands to execute regardless of the settings for the `read_only` and `super_read_only` system variables and the `autocommit` mode. (Bug #26414532, Bug #86224)

- **Replication:** The binary log function `MYSQL_BIN_LOG::new_file_impl` returned the error “Can't open file” (`ER_CANT_OPEN_FILE`) when it should have returned “Error writing file” (`ER_ERROR_ON_WRITE`). (Bug #26370868, Bug #86870)

- **Replication:** When write sets are used for parallelization by a replication slave (as specified by the `binlog_transaction_dependency_tracking` system variable), the case and accent sensitivity of the database are now taken into account when generating the write set information. Write set information is generated when the `transaction_write_set_extraction` system variable is enabled. Previously, duplicate keys could be incorrectly identified as different, causing transactions to have incorrect dependencies and so potentially be executed in the wrong order. (Bug #26277771, Bug #86078)

- **Replication:** When the `transaction_write_set_extraction` option was enabled, there was a risk of unnecessary serialization while foreign keys were gathered if concurrent DDL took place. Group Replication now takes advantage of the new Data Dictionary to interact with table definitions and foreign keys, which has solved this potential serialization. (Bug #26187850)

- **Replication:** The receiver thread for semisynchronous replication was not able to receive acknowledgements from slaves that used compression of the master/slave protocol (`slave_compressed_protocol=ON`). The receiver thread now handles compressed acknowledgements correctly. (Bug #26027024, Bug #86230)
• **Replication**: The `mysql_reset_connection()` function now clears the write set session history. (Bug #25950554, Bug #86063)

• **Replication**: On replication slaves, in the `XA_STATE` field in the Performance Schema table `events_transactions_current`, the state of XA transactions was incorrectly reported as `COMMITTED` instead of `PREPARED` after the `XA PREPARE` statement was applied on the slave. (Bug #25940184)

• **Replication**: In a multi-source replication topology, a memory leak could occur on the slave when `binlog_rows_query_log_events` was enabled on the master, and a statement already applied from another channel was skipped on the slave. In this situation, the instance of the `Rows_query` log event stored on the slave was not being deleted. The log event instance is now cleaned up and the memory is freed. Thanks to Vlad Lesin for his contribution to the patch. (Bug #25695434, Bug #85371, Bug #85034)

• **Replication**: Queries to the Performance Schema `replication_applier_global_filters` and `replication_applier_filters` tables, which show the global and channel-specific replication filters configured on a replication slave, have been optimized so that a view is generated only when the filters are changed. Previously, a view was generated for every row that was created. (Bug #25694140)

• **Replication**: A memory leak was fixed in GTID-based replication. Memory was not being freed after the repository tables were updated for skipped or ignored events. (Bug #25656123, Bug #85251)

• **Replication**: When a worker thread on a multithreaded slave failed to apply a transaction on which a later transaction depended, the coordinator thread could begin scheduling the dependent transaction before being notified of the issue. If a `STOP SLAVE` request was made during this situation, it caused an assertion to be raised in debug builds. (Bug #25585436)

• **Replication**: When `group_replication_enforce_update_everywhere_checks=ON` the Group Replication plugin checks if there are foreign key cascades and disallows updates to such tables. However `SET NULL` operations were not being checked, which could cause data inconsistency. Now, when `group_replication_enforce_update_everywhere_checks=ON`, operations on child tables are blocked if the table has a `SET NULL` option configured. (Bug #25404162)

• **Replication**: On Windows, errors generated by Group Replication now contain the detailed error message rather than just the error number. (Bug #24918678)

• **Replication**: With statement-based replication in use, if an `UPDATE` or `DELETE` statement was used inside an XA transaction ending with `XA COMMIT ONE PHASE`, and the statement did not affect any rows, a replication error occurred. An `XA END` statement was not written to the binary log, so slave servers identified the XA transaction as still being active at the time of the commit request. The required `XA END` statement is now written even if the transaction affected no rows. (Bug #24812958, Bug #83295)

• **Replication**: Replication clients no longer enable `LOCAL` capability for `LOAD DATA` statements, because they do not use `LOAD DATA LOCAL` statements. (Bug #24763131)

• **Replication**: The behavior of mixed-format replication (`binlog_format=MIXED`) has changed with regards to temporary tables. Previously, when mixed-format binary logging was in use, if a statement was logged by row and the session that executed the statement had any temporary tables, all subsequent statements were treated as unsafe and logged in row-based format until all temporary tables in use by that session were dropped. Also, on a replication slave with `log_slave_updates` enabled, row-based logging was incorrectly continued across all subsequent sessions for the duration of the connection, as reported in the bug.

Now, when mixed binary logging format is in use, statements that exclusively use temporary tables are not logged. Statements that involve a mix of temporary and non-temporary tables are logged on the master only for the operations on non-temporary tables, and the operations on temporary tables are
not logged. The exception is if the creation of a temporary table was recorded in the binary log using statement-based format. In this case, a `DROP TEMPORARY TABLE IF EXISTS` statement is logged on the master when the temporary table is dropped.

With this change in behavior, the remaining statements in the session that do not involve temporary tables no longer need to be treated as unsafe. The safe statements are now logged in statement-based format, and the unsafe statements are logged in row-based format, according to the normal behavior for mixed format replication, regardless of the presence of temporary tables in the session. Also, the issue reported in the bug has been fixed so that subsequent sessions using the connection now use the appropriate logging format for the session, regardless of the format used by earlier sessions.

When `binlog_format` is `ROW` or `STATEMENT`, the behavior remains as before. For row-based binary logging format, operations on temporary tables are not logged, with the exception of the `DROP TEMPORARY TABLE IF EXISTS` statement as for mixed format. For statement-based binary logging format, operations on temporary tables are logged on the master and replicated on the slave, provided that the statements involving temporary tables can be logged safely using statement-based format. `binlog_format=STATEMENT` is now the only logging mode in which temporary tables are replicated on the slave.

You cannot now change the `binlog_format` setting from `ROW` or `MIXED` to `STATEMENT` at runtime, because any `CREATE TEMPORARY TABLE` statements will have been omitted from the binary log in the previous mode. You can still switch from `STATEMENT` to `ROW` or `MIXED` format, even when temporary tables have been created.

Thanks to George Lorch and Laurynas Biveinis from Percona for the patch. (Bug #18843730, Bug #72475)

- **Replication:** All servers that belong to a group must have unique UUIDs set by `server_uuid`, but this was not being enforced by Group Replication and it was possible to add members with duplicated UUIDs. (Bug #88452, Bug #27105803)

- **Replication:** During distributed recovery as part of joining the group, when the applier was signaling that it had applied all transactions, it was also blindly searching for partial transactions. This was to avoid future applier errors, which would happen if the applier stopped at this point. However, this search and remove only made sense for applier stop cases. Upon execution completeness it should not be done, otherwise it can corrupt or purge the applier relay log, which can lead to data loss. To solve this issue, when the applier is waiting for execution completeness, it no longer searches for and removes partial transactions. (Bug #88304, Bug #27049034)

- **Replication:** Group Replication executes internal operations on the server during start and stop of the plugin, such as enabling or disabling read only mode, using an internal session. When this internal session was opened, if the total number of sessions exceeded the number of permitted open sessions set by `max_connections`, the operation was failing as expected but a thread was left behind, which later would cause issues. (Bug #88182, Bug #27008102, Bug #27016552)

- **Replication:** If Group Replication was configured to start on server boot when the server was being initialized using `--initialize` or `--initialize-insecure`, because the replication applier infrastructure was not initialized this resulted in an assertion. Now, Group Replication is not started when the server is being initialized. (Bug #87759, Bug #26802395)

- **Replication:** In a group with heavy load, joining members could need to retrieve a large amount of data to gain synchrony with the group. If the amount of data retrieved exceeded the `View_change` packet size of 4Mb the members would fail to join the group and enter Error state. Now, the packet size is taken from `slave_max_allowed_packet`, which defaults to 1GB. Depending on the load your group processes, you might want to increase the packet size further by configuring `slave_max_allowed_packet`. (Bug #87701, Bug #26770576)
MySQL 8.0 Release Notes

- **Replication:** In a group where a joining member consistently received transactions, the joining member could sometimes not enter the online state. This was due to the way the incoming queue of messages was tested. (Bug #87631, Bug #26731317)

- **Replication:** Changes to Group Replication variables while starting or stopping the plugin were not being correctly validated. Now, the variables can only be changed if the plugin is not changing state. (Bug #86874, Bug #26372117)

- **Replication:** Regardless of the number of virtual IPs configured on a machine, Group Replication could access only the first 12 addresses. (Bug #86772, Bug #26324852)

- **Replication:** The delayed initialization mechanism used for server starts has been improved. Now, it only blocks connections until the server is in read mode. (Bug #86271, Bug #26037344)

  References: See also: Bug #84731, Bug #25475132.

- **Replication:** When a primary member, for example the primary in single-primary group or in a multi-primary group, which also had asynchronous replication channels feeding data into it was stopped, the asynchronous channels would continue applying changes. Although `super_read_only` was being set when `STOP GROUP_REPLICATION` was issued, this did not stop any running asynchronous replication channels which were already running. This meant that changes could be made locally on the member, and that the asynchronous replication channels had to be stopped manually. Now when Group Replication stops, either due to an error or when `STOP GROUP_REPLICATION` is issued, all asynchronous replication channels are stopped. (Bug #86222, Bug #26024253)

- **Replication:** The logging of Group Replication has been improved. Now logging includes information when a member joins or leaves, when the view changes, and so on. (Bug #84798, Bug #25495393)

  References: See also: Bug #26422857.

- **Linux:** On Alpine Linux, `mysql` would lose its connection to the server if its standard output was not writable. Also, for `mysql` and `mysqldump`, order of result flushing for `stdout` and `stderr` is now deterministic. (Bug #27169809)

  References: See also: Bug #17583.

- **Microsoft Windows:** On Windows, with the `myisam_use_mmap` and `flush` system variables enabled, MyISAM did not always flush table files properly. (Bug #26880757)

- **JSON:** JSON expressions used as arguments with the `LAG()` function were not always evaluated correctly. (Bug #26740557)

- **JSON:** Repeated execution of a prepared statement that employed `JSON_ARRAY()` was not handled correctly. (Bug #26704312)

  References: This issue is a regression of: Bug #25867454.

- **JSON:** When executing the `JSON_INSERT()` function, the check that is performed to determine whether or not a given insert is being made into the root element tested whether the length of the path was 1—that is, whether the path consisted of a single leg determining which position the inserted element has inside the root element. A problem occurred when there were auto-wrapping path legs at the beginning of the path, in which case a path whose length is greater than 1 might also refer to an element in the root, so that checking the path length did not reliably inform us whether the target element of the insert was the root or some other element.

  To fix this, the check of the path length for detection of the root element has been replaced with a check as to whether the matched element has a parent; if it has none, it must be the root element. (Bug #26649978)
MySQL 8.0 Release Notes

References: This issue is a regression of: Bug #86213, Bug #26022576.

• **JSON:** When serializing a JSON value to its binary representation, it is necessary to make sure that the destination buffer has sufficient space to hold an integer or double value of the required size. Allocation of this buffer previously reserved only the minimum amount of memory needed, which made it very likely that a reallocation would be needed shortly thereafter. This could adversely affect performance, especially when serializing arrays with many numeric values. The serialization is now performed in a manner such that the allocation increases the size of the destination buffer size exponentially, which reduces the amount of the time spent performing this task when processing large arrays. (Bug #88656, Bug #27171283)

• **JSON:** When a JSON document was converted to string representation, floating-point values that had no fractional part could be represented such that they became indistinguishable from integers. When the string representation of such a JSON document was passed through the JSON parser again, the information that the numeric value was originally specified as a floating-point value was lost.

To rectify this problem, a fractional part is now added to the string representation of a floating-point value in a JSON document if the value has no fractional part and is not represented using scientific format. This makes the string representation of a floating-point value distinguishable from that of an integer, so that it continues to be treated as a floating-point number even if the string is parsed again.

This fix also makes `ST_GeomFromGeoJSON()` use the same JSON parser as the other JSON functions rather than its own custom parser as had been the case since MySQL 5.7.8; this special handling was due to the fact that `ST_AsGeoJSON()` dropped the fractional part of negative zero (-0 instead of -0.0), causing the JSON parser to interpret -0 as integer 0, thus losing the distinction between positive and negative zero. Since `ST_AsGeoJSON()` now uses the standard JSON parser, it represents negative zero as correctly as -0.0, obviating any need for `ST_GeomFromGeoJSON()` to preserve negative zero explicitly on its own when parsing the output from `ST_AsGeoJSON()`. (Bug #88230, Bug #27028889)

References: See also: Bug #19504183.

• **JSON:** When inserting JSON values created from the result of a GROUP BY query, the inserted values could sometimes include the concatenation of all the values previously inserted into that column. (Bug #87854, Bug #26867509)

• **JSON:** When called using strings extracted from JSON documents as arguments, the `LEAD()` and `LAG()` functions returned the same value for every row. (Bug #87839, Bug #26848089)

• **JSON:** The microseconds part of the last-updated field in each histogram in the `INFORMATION_SCHEMA.COLUMN_STATISTICS` table (showing when the histogram was last updated) was dropped when serializing the histogram into JSON and so was not stored. (Bug #87833, Bug #26846289)

• **JSON:** When a `JSON_SET()` statement updated a JSON value to the same value using a partial update (in other words, when the partial update was essentially a NOOP), it was possible that logical diffs for this operation were produced, even though no binary diffs were produced. Now in such cases, neither logical update nor binary diffs are generated. (Bug #87113, Bug #26483625)

• **JSON:** Following the implementation of JSON partial updates, the same JSON document could have different binary representations on the master and the slave. This could lead row-based replication—which uses binary equality to find the matching row on the slave—to fail if this occurred. Now the string representation of the JSON document is used for the comparison instead.

Also as a result of this fix, updates can be skipped in more cases than previously; this is true where the binary representation has changed, but not the contents of the document. (Bug #86532, Bug #26177130)
• Ubuntu 14.04 and Debian 8 are no longer supported. (Bug #27422291)

• For builds on 32-bit platforms with Undefined Behavior Sanitizer enabled, a stack-overrun check could cause a server exit. (Bug #27224961)

• The server could hang during spatial reference system (SRS) creation while another session was using that SRS. (Bug #27220467)

• A lock for the privilege cache was acquired unnecessarily during privilege-checking operations not involving table permissions. (Bug #27197483)

• Persisting the read-only `gtid_owned` or `gtid_executed` system variable caused an assertion failure at server startup. These variables can no longer be persisted. (Bug #27193853)

• Improper handling of plugin loading and unloading could cause a server exit. (Bug #27151550, Bug #88589, Bug #27116827, Bug #88483)

• Error propagation from some windowing functions was not always performed correctly. (Bug #27135084, Bug #27136492)

• Negation of some very large values was not handled correctly by an internal function. (Bug #27134168)

• Instituted stricter checks when performing addition involving date intervals. (Bug #27134148)

• Recently introduced SRID and COLUMN_STATISTICS metadata locks were not instrumented by the Performance Schema. (Bug #27124506)

• The name of a derived table was not saved before the table was materialized and assigned the name of the temporary table. Later, when trying to reset the table name, this caused the server to fail due to the missing reference to the original value of the name. (Bug #27121663)

• Performance Schema queries that used indexes on `OBJECT_TYPE` columns could return incorrect results. (Bug #27121500)

• Compiling with `-DWITH_ASAN=1` and `-DWITH_ASAN_SCOPE=1` detected a stack-use-after-scope memory error. (Bug #27108794, Bug #88460)

• `FILE` privilege checking for prepared `SELECT ... INTO OUTFILE` statements was incorrect. (Bug #27094955)

• Some messages with information priority were written to the error log when `log_error_verbosity` was less than 3. (Bug #27082862)

• The `ha_create_table_from_engine` function failed to pass a table object to the `ha_create` routine. (Bug #27066335)

• Partition by and order by elements of unused window definitions were not included when estimating memory requirements. This is resolved by assigning a parsing context (`CTX_WINDOW`), but only when the current context is `CTX_NONE`. As part of this fix, unused window definitions are now removed after being checked for syntax and semantic errors. (Bug #27062031)

• `GROUP BY` with a `ROLLUP` that generated `NULL` was not handled correctly. (Bug #27060420)

• An Event Scheduler event for which global autocommit was disabled at event expiration time caused an assertion to be raised. (Bug #27041552, Bug #88255)

• Length calculations for string-valued user-defined variables could be incorrect if the collation was changed. (Bug #27041543, Bug #88256)
• **CREATE TABLE ... SELECT** statements that attempted to create a non-**InnoDB** table raised an assertion if a pre-existing view referenced the table to be created. (Bug #27041536, Bug #88258)

• When used as an argument to the **IF()** function, the value of a **TIMESTAMP** column could be handled differently for different collations. (Bug #27041526, Bug #88259)

• With statement-based binary logging, using **CREATE TABLE ... SELECT** to create a **BLACKHOLE** table caused an assertion to be raised. (Bug #27041516, Bug #88260)

• For debug builds, a **TIMESTAMP**-related assertion could be raised with **explicit_defaults_for_timestamp** enabled. (Bug #27041502, Bug #88261)

• Under **LOCK TABLES**, an attempt to execute a DML statement on a table with foreign keys led to assertion failure if the statement was incompatible with the mode under which the tables in the foreign key relationship were locked. (Bug #27041477, Bug #88264)

• With a **LOCK TABLES** statement active, queries that select from the **INFORMATION_SCHEMA.FILES** table could raise an assertion trying to obtain a metadata lock. (Bug #27041452, Bug #88266)

• Server component installation did not properly perform auto-increment handling, which could result in a server exit. (Bug #27041374, Bug #88276)

• With **big_tables=1** and **character_set_connection=ucs2**, **SHOW TABLE STATUS** could raise an assertion. (Bug #27041323, Bug #88279)

• With a backup lock active, removal of binary log files and relay log files incorrectly was permitted. (Bug #27030339, Bug #88238)

• When evaluating an end-range condition in a scan of a non-covering secondary index, and the end-range condition referenced an indexed virtual column, **InnoDB** sometimes read the wrong column from the index, which could lead to assertion failures or wrong results. (Bug #27010089)

• In **event** items in filter rules, the **audit_log** plugin did not properly process values specified as a **JSON** array. (Bug #27010045)

• An integer overflow was sometimes possible when adding date values, with the potential to return invalid results. (Bug #27004806)

• **TRUNCATE TABLE** on a **InnoDB** table with an active **LOCK TABLES** raised an assertion if more than one table instance was locked. (Bug #26999526)

• Some windowing functions including **NTH_VALUE()** did not return **NULL** in all cases in which they should have. (Bug #26975882)

• If one of the temporal arguments to **LEAST()** or **GREATEST()** was of type **DATETIME**, but the aggregated data type was something other than **VARCHAR** or a temporal type, the **LEAST()** or **GREATEST()** item had temporal properties set for it in spite of the fact that it was not of a temporal data type. The temporal properties for the **LEAST()** or **GREATEST()** item, including its fractional precision, were therefore not calculated correctly, leading to failure at a later point in statement execution.

To fix this issue, the data type of **LEAST()** or **GREATEST()** is now temporarily set to a temporal type if one or more, but not all, of its arguments are of temporal types regardless of their aggregated data type. (Bug #26975864)

• **IFNULL()** did not always check for errors correctly when processing multiple arguments. (Bug #26975848, Bug #27062796, Bug #27062694)
MySQL 8.0 Release Notes

- With compression enabled for the client/server protocol, logical packets that are multiples of size 0xFFFFFF could cause the connection to be dropped. Thanks to Facebook Inc. for the patch. (Bug #26974113, Bug #88092)
- Installing and uninstalling a plugin many times from multiple sessions could cause the server to become unresponsive. (Bug #26946491)
- A virtual column definition that included a function removed in MySQL 8.0 caused startup to fail with a “No database selected” error when starting the MySQL 8.0 server on a MySQL 5.7 data directory. (Bug #26945125, Bug #88040)
- These errors occurred after an in-place upgrade from MySQL 5.7 to 8.0:
  - Starting the server with an `--explicit-defaults-for-timestamp=0` setting returned an Invalid default value for 'cached_time' error.
  - Starting the server with `--initialize` and `--explicit-defaults-for-timestamp=0` configuration settings returned an Invalid default value for 'SET_TIME' error.
  - Duplicate SET data type values caused a Duplicated value in SET error, regardless of the sql_mode configuration setting.
    (Bug #26944731, Bug #88039, Bug #26948678, Bug #88032)
- Queries with a common table expression and a derived table or view that contained a window function produced incorrect results. (Bug #26907753, Bug #87939)
- The deprecation warnings were clarified regarding use of the `--symbolic-links` and `--skip-symbolic-links` server options (and their equivalents). The server no longer warns about a missing data directory when invoked with the `--help` option. (Bug #26898576, Bug #87913)
- `VALUES()` was not handled correctly in some cases. (Bug #26881946)

References: See also: Bug #19601973, Bug #17458914.

- The tablespace discovery mechanism in MySQL Cluster was disabled, which prevented serialized dictionary information from being imported into the data dictionary in cases where the table exists in the storage engine dictionary but not in the MySQL data dictionary. (Bug #26867488)
- For debug builds, validation checks on relevant generated columns could be missed for UPDATE statements, leading to a server exit. (Bug #26838771)
- For window functions, an error could be produced that a window was not defined, when it was defined. (Bug #26813454, Bug #87780)
- When a materialized derived table was determined to be superfluous, the routine that deleted the corresponding object left it in an inconsistent state. Now in such cases, the derived table's TABLE_LIST object left in a consistent state after its materialized object is deleted, by setting its table pointer to NULL. (Bug #26798989)
- Incorrect results were obtained for a query with MAX() and a HAVING clause used inside a view. (Bug #26781725)
- Metadata locking for definition changes to tables underlying a view could be inconsistent with metadata locking for other statements on the tables. (Bug #26770836)
- In MySQL 8.0, view column names are restricted to 64 characters. A MySQL 5.7 view with longer column names was marked as invalid during an upgrade to 8.0 was marked invalid. Now such views
produce an error during the upgrade and must be altered to have legal column names before the
upgrade will succeed. (Bug #26743291, Bug #87650)

• Concurrent calls to `GET_LOCK()` could cause deadlock, even with a wait time of 0. (Bug #26739438, Bug #87620)

• `FROM_UNIXTIME()` did not always work correctly with `LAG()`. (Bug #26739028)

• Following an `INSERT` statement with `BLOB` values in the `ON DUPLICATE KEY UPDATE` clause that failed with a constraint violation, a similar statement with no reason to return an error could cause a server exit. (Bug #26734162)

• The Performance Schema now stores rewritten rather than raw SQL statement text when available. (Bug #26732229)

• Re-executing a prepared statement that used window functions could cause a server exit. (Bug #26730020)

• An in-place upgrade to MySQL 8.0 caused a server exit if tables contained columns with a pre-5.0 `DECIMAL` data type. This data type is not supported, so upgrades now detect such columns and warn that in-place upgrade cannot be done. Affected tables must be upgraded by dumping and restoring them. (Bug #26727481)

• A comparison operator used to order keys in a data dictionary cache hash map incorrectly determined that two storage-engine private IDs used by different storage engines were equal. (Bug #26723442)

• Values in the `XID_GTRID` column of the Performance Schema `events_transactions_current` table were displayed incorrectly for XA transactions. (Bug #26710081, Bug #87559)

• Incorrect results or a server exit could result when `SHA2()` was passed a user-defined variable in some character sets. (Bug #26704451)

• Incorrect NULL handling by `LAG()` and `LEAD()` could cause a server exit. (Bug #26703246, Bug #26703156)

• `mysqlpump` no longer includes the `SQL_NO_CACHE` modifier in statements because that modifier is now deprecated and results in deprecation warnings. (Bug #26694675)

• If an error occurred while setting up the temporary table for duplicate weedout in a semi-join (for example, because the disk was full), the server did not terminate gracefully with an appropriate error message.

Now in the event that the temporary table is not successfully created, the query is aborted. (Bug #26679983)

• The server did not always clean up correctly after executing an `IN` subquery that used a hash semi-join. (Bug #26679495)

• Building with the `--WITHOUT_SERVER=ON` CMake option failed due to attempting to link the `authentication_ldap_sasl_client` client-side plugin against the embedded server library. (Bug #26665217)

• During data directory creation or upgrade from MySQL 5.7 to 8.0, server startup would fail due to a Performance Schema initialization failure if the server was started in `read_only` mode. Additionally, Information Schema metadata was not updated at startup, and Performance Schema and Information Schema version information was stored without verifying that schema tables were created.

The versioning scheme used for the data dictionary and Information Schema system views is now consistent with the Performance Schema versioning scheme. (Bug #26636238, Bug #87436)
• The **CREATE_OPTIONS** column in the **INFORMATION_SCHEMA.TABLES** table did not show correct information. (Bug #26634507)

• Incorrect results could be returned for queries that used an outer join and a derived table referenced a **const** value from an inner table of the outer join. (Bug #26627181)

• **AFTER UPDATE** triggers were not invoked for **INSERT ... ON DUPLICATE KEY UPDATE** when the value to be updated and the new value were the same. (Bug #26626277, Bug #87371)

• Assignment of anonymous roles to the **mandatory_roles** system variable was incorrectly permitted. Additionally, assigning a value to **mandatory_roles** now requires the **ROLE_ADMIN** privilege, in addition to the **SYSTEM_VARIABLES_ADMIN** or **SUPER** privilege normally required to set a global system variable. (Bug #26576989)

• The server fell back to using the built-in error messages if the **lc_messages_dir** value was invalid at server startup, but not if **lc_messages** or **lc_time_names** were invalid. Now the server uses the built-in messages if any of those variables are invalid at startup. (Bug #26576922)

• **SET DEFAULT ROLE ALL** did not include roles named in the **mandatory_roles** system variable. (Bug #26571995)

• On Windows, **CMake** did not automatically add x64 toolchain support for some Visual Studio versions. (Bug #26566360)

• Problems could occur when a derived table with an **ORDER BY** clause was merged into an outer query, and when the columns from the **ORDER BY** were not also referenced in the outer query. (Bug #26542829)

• Parallel inserts of schema SDI into the SDI B-tree could raise an assertion when creating tables in the same schema in parallel. (Bug #26539665, Bug #87225)

• Changing the **UMASK** and **UMASK_DIR** environment variables from their default values had no effect on database directory and table file access. (Bug #26529942)

• For debug builds, incorrect nullability assessment of derived table column references could cause **CONCAT()** to raise an assertion. (Bug #26524721)

• A server exit could result from simultaneous attempts by multiple threads to register and deregister metadata Performance Schema objects, or to acquire and release metadata locks. (Bug #26502135)

• MSI packages for Windows failed to detect when Microsoft Visual C++ 2010 Redistributable Package was installed. (Bug #26501092, Bug #87139)

• Queries that used window functions for tables that contained a **BLOB** column could cause a server exit. (Bug #26496880)

• Persisted variables belonging to plugins were not always handled properly at server startup. (Bug #26495619)

• A server exit could occur for queries that used **DISTINCT** and **ORDER BY ... LIMIT** and were executed using range access and a temporary table for the **ORDER BY**. (Bug #26483909)

• LDAP authentication plugins could fail if their associated system variables were set to invalid values. (Bug #26474964)

• The Linux RPM spec file for RHEL6 and higher is updated with comments that recommend installing the **redhat-rpm-config** package to add convenience macros that make rebuilding the RPM package easier. Thanks to Simon Mudd for the patch. (Bug #26474153, Bug #87098)
MySQL 8.0 Release Notes

• If the error log was misconfigured and the server could not start, no output describing the problem was produced. (Bug #26447825, Bug #87087)

• When a materialized semi-join operation was evaluated more than once, and one of the tables in the materialization was a `const` table (that is, with join type `JT_CONST`), invalid data was accessed during the second materialization when referencing the `const` table. (Bug #26436185)

• Password-expiration options did not work correctly for authentication plugins that use external authentication methods. (Bug #26435766)

• Adding an `ORDER BY` to a query that included an outer join and a subquery caused a constant value defined for a column in the subquery to be incorrectly promoted to a constant value in the case when the subquery returns 0 rows. (Bug #26432173)

• For the `autocommit` system variable, the Performance Schema `variables_info` table always reported the `VARIABLE_SOURCE` column as `COMPILED`. (Bug #26428017)

• For debug builds, `INSERT IGNORE` statements that tried to insert `NULL` into a `GEOMETRY NOT NULL` column raised an assertion because there is no valid value to convert the `NULL` to. This is now handled as a nonignorable `ER_BAD_NULL_ERROR_NOT_IGNORED` error. (Bug #26412713)

• `SET PERSIST_ONLY` changed the `VARIABLE_SOURCE` column of the Performance Schema `variables_info` table when it should not have. (Bug #26395134)

• The server failed to check the maximum path length for partition names. (Bug #26390632)

• Problems occurred when a window with buffering followed an equi-join on a unique index, due to the fact that the window modified the input record with the assumption that, the next time control passes back to the join, a new record was read to the input record. This problem is addressed by reinstating the input record in such cases. (Bug #26389508)

• Identifiers containing a mix of backslashes and backticks could be parsed incorrectly. (Bug #26372491)

• `audit_log` plugin THD objects could be created with incorrect thread ID information, leading to assertion failure. (Bug #26362452)

• The `HISTOGRAM` column in the `column_statistics` data dictionary table used a key named `charset-id` to indicate collation numbers. This key has been renamed to `collation-id`. (Bug #26330090, Bug #86794)

• Starting the server with the `--skip-name-resolve` option could cause `localhost` in account entries to match non-`localhost` hosts. (Bug #26328274, Bug #26202411, Bug #86546)

• if a configured error log service existed but could not be initialized, log information was lost. Now if a log service is configured at startup but cannot be initialized, diagnostics about the problem are sent to the default log service (or if that fails, directly to the error stream), and the server exits. If a log service is configured at runtime but cannot be initialized, diagnostics are sent to the client. (Bug #26286871, Bug #86728)

• Installing and uninstalling Performance Schema example plugins concurrently with deletes from tables associated with those plugins could cause a server exit. (Bug #26281359)

• When `HASH_SCAN` was specified as one of the values for the `slave_rows_search_algorithms` system variable, which is the default from MySQL 8.0.2, and row-based replication was in effect, updates to a table containing virtual generated fields could raise an assertion. The issue was caused by an error when generating string representations of the virtual generated fields in order to create hashes for use in searches. To remove the issue, MySQL no longer creates hashes for virtual generated fields. (Bug #26280724)
• The server permitted `SHOW CREATE TABLE` on nontable files created for full-text searching. (Bug #26271244)

• For debug builds, `CREATE OR REPLACE VIEW` for an existing view raised an assertion for column names greater than 64 characters. Now an appropriate error is reported. (Bug #26267689)

• `SET PERSIST` could be ineffective due to sorting variables written to `mysqld-auto.cnf`. Variables are now written in the order persisted. (Bug #26259671)

• Attempting a partial backup with `mysqlpump` on a GTID-enabled server failed and produced an error message suggesting incorrectly that this was not possible. (It is possible using the `--set-gtid-purged` option.) (Bug #26199978)

• `GRANT GRANT OPTION ON *.* TO user` granted `GRANT OPTION` for static but not dynamic privileges. `REVOKE ALL ... FROM CURRENT_USER()` revoked static but not dynamic privileges. (Bug #26191109, Bug #25658967)

• Error logging could attempt to log freed messages, resulting in a server exit. (Bug #26188656, Bug #86562)

• A `HAVING` condition was optimized away for an alias on an aggregate column where there was no `GROUP BY` clause. (Bug #26188578)

• `ST_Crosses()` could return an incorrect result when at least one parameter is a geometry collection and multiple elements of the geometry collection must be taken into account in order to determine whether the geometries cross. (Bug #26188208, Bug #86559)

• `MBROverlaps()` incorrectly returned false for two crossing perpendicular lines. (Bug #26188118, Bug #86558)

• `ALTER USER user DEFAULT ROLE ALL` produced an error. (Bug #26174169)

• `mysqldump` exited abnormally for large `--where` option values. (Bug #26171967, Bug #86496, Bug #27510150)

• A query using a window function with a window which partitioned or ordered on the result of an aggregate function where this evaluated as `NULL` returned incorrect results. (Bug #26164633)

• The Performance Schema could leak memory due to nondeletion of file instances created for `ALTER TABLE` operations that used the table-copy algorithm. (Bug #26152751, Bug #86482)

• Failed creation of a temporary table using `REPLACE` could under some circumstances result in an assertion in a later statement. (Bug #26126789, Bug #86422)

• `mysqlpump` did not properly parse `TABLESPACE` clauses in the result from `SHOW CREATE TABLE` statements it executed to determine table structure. (Bug #26116415)

• The binary file for the `udf_example` user-defined function was omitted from binary distributions. (Bug #26115002, Bug #29178542)

• Long `SET PERSIST` statements could cause a server exit. (Bug #26100122)

• An incorrect formula was used to calculate maximum length of result strings for a few string functions: `QUOTE()`, `AES_DECRYPT()`, and `WEIGHT_STRING()`. This could affect, for example, the length of character columns created for `CREATE TABLE ... AS SELECT ... QUOTE()`. (Bug #26049942, Bug #86305)

• Schema creation and removal operations could fail due to checking for schema directories under the data directory rather than checking the data dictionary. (Bug #26043994, Bug #86282)
• **SHOW PLUGINS** did not handle plugins that were terminating, resulting in a server exit. The statement now displays the status for such plugins as **DELETING**. (Bug #26029765, Bug #86243)

• Some statements could cause a buffer overflow in the digest code. Thanks to Laurynas Biveinis and Roel van de Paar for the patch. (Bug #26021187)

• Previously, when the Performance Schema failed to initialize, it wrote a nonspecific `init failed` warning to the error log. Now it prints more specific messages about which memory allocation failed. (Bug #25996291)

• Incorrect results could occur on a table with a unique index when the optimizer chose a Loose Index Scan even though the unique index had no index extensions. (Bug #25989915, Bug #86165, Bug #26532061, Bug #87207)

References: This issue is a regression of: Bug #21749123, Bug #78244.

• For **XA COMMIT**, precommit handling could set an error in the diagnostics area that was not reported correctly on the calling side, causing an assertion to be raised. (Bug #25978684, Bug #86142)

• The **MIN_VALUE** column of the Performance Schema `variables_info` table displayed incorrect values on 32-bit big-endian platforms. (Bug #25967079)

• A memory leak occurred when the optimizer excluded a subquery associated with a temporary table. (Bug #25951134)

• An assertion could be raised for updates when a view or derived table considered read only had nested references not seen as read only. (Bug #25832861, Bug #85796)

• Certificate and key files automatically generated by the server could have an incorrect access mode. (Bug #25832856)

• Queries on the **INFORMATION_SCHEMA TABLES** and **STATISTICS** tables, if evaluated using Index Condition Pushdown, could push down internal data dictionary functions, resulting in an assertion being raised. (Bug #25820175, Bug #85765)

• **ST_AsText()** could read freed memory. (Bug #25818451)

• **CREATE USER IF NOT EXISTS** was not written to the binary log if the user existed. This could result in inconsistent replication behavior if the user did not exist on slave servers. A similar issue occurred for **ALTER USER IF EXISTS**. To avoid inconsistencies, these statements now are written to the binary log. (Bug #25813089, Bug #85733)

• An invalid **utf8** input string caused a heap buffer overflow error. (Bug #25811623, Bug #25946444)

• A race condition made it possible to cause a server exit by persisting variables from multiple sessions simultaneously. (Bug #25768813)

• Plugins can create or drop **INFORMATION_SCHEMA** tables, but views that reference **INFORMATION_SCHEMA** tables were not validated when plugins were unloaded or unloaded. (Bug #25761992, Bug #85579)

• **mysql** wrote some password-related statements to the `.mysql_history` file. (Bug #25750609)

• Incorrect handling of internal memory buffers could cause a server exit. (Bug #25737271)

• On a read-only server with GTIDs enabled, a **DROP TEMPORARY TABLE IF EXISTS** statement relating to a nonexistent or filtered table could write an unnecessary transaction to the binary log and create an unnecessary GTID. In this situation, the missing temporary table was previously assumed to be
MySQL 8.0 Release Notes

transactional, leading to the statement being split. Now, MySQL checks that the temporary table exists and that its `DROP TEMPORARY TABLE` statement is recorded in the binary log. If this is not the case, no GTID is created. Thanks to Laurynas Biveinis for the patch. (Bug #25656992, Bug #85258)

- The MeCab full-text parser plugin failed to load on Windows. (Bug #25633175)
- The `SET_TIME` column of the Performance Schema `variables_info` table was initialized incorrectly. (Bug #25608115)
- Executing a stored procedure containing a statement that created a table from the contents of certain `SELECT` statements could result in a memory leak. (Bug #25586773)
- The Performance Schema failed to check the maximum host length for client connections. (Bug #25510805)
- For spatial functions, some set operations produced a result with SRID 0 when given arguments in a different SRID. (Bug #25510403)
- Large `--ssl-cipher` values could cause client programs to exit. (Bug #25483593)
- A missing argument-count check during preparation of a stored procedure call could result in a server exit. (Bug #25398451, Bug #84512)
- Temporary tables used in processing a recursive common table expression with `UNION DISTINCT` and a great many columns now use the `MEMORY` engine instead of `InnoDB`. (Bug #25190109)
- If the MySQL root user account was renamed, a query that accessed an `INFORMATION_SCHEMA` view returned an error stating that the user specified as the definer does not exist. To avoid this error, a new system user account, 'mysql.infoschema'@'localhost', is now the DEFINER for `INFORMATION_SCHEMA` views. (Bug #25185947, Bug #84027)
- When an `UPDATE` required a temporary table having a primary key larger than 1024 bytes and that table was created using `InnoDB`, the server could exit. (Bug #25153670)
- `SET DEFAULT ROLE` was not transactional like other account-management statements. (Bug #25122897)
- `mysqlpump` included the `gtid_executed` table in dumps of the `mysql` system database, causing the `gtid_executed` position to be lost upon server restart after the dump was reloaded. `mysqlpump` no longer dumps the `gtid_executed` table. (Bug #25109007)
- For geometry calculations, invalid input parameters could lead to an incorrect result buffer and cause an assertion to be raised or a server exit. (Bug #25062396)
- `IFNULL(decimal, int)` could lose a digit after the decimal point when used in a query that included `GROUP BY` and was executed using a temporary table. (Bug #25051195, Bug #83699)
- For some queries, such as those involving `UNION`, column width for `GROUP_CONCAT()` could be calculated incorrectly, leading to incorrect application of `group_concat_max_len`. (Bug #25050090, Bug #83667)
- Audit logging of events for the Performance Schema `global_variables` table was improved so as to not report events for rows materialized but not reported to the SQL layer. (Bug #24970428)
- `ST_Buffer()` could return an invalid result or raise an error for some inputs that should have produced a valid output geometry. (Bug #24947868, Bug #26735293, Bug #25662426)
- For builds with AddressSanitizer or Undefined Behavior Sanitizer enabled, division by zero could occur during Performance Schema timer initialization. (Bug #24785784)
• Operations that rely heavily on the metadata locking (MDL) subsystem caused a performance degradation. Traversal of MDL ticket lists was time consuming in cases where there were large number of MDL tickets. (Bug #24734971, Bug #83143)

• When `binlog_format` is `ROW` or `MIXED`, operations on temporary tables are not logged. Previously, the exception to this rule was that when the connection was terminated at the end of the session, the statement `DROP TEMPORARY TABLE IF EXISTS` was logged for any temporary tables that had been opened in the session. For row-based replication, this behavior caused an unnecessary write to the binary log, and added a transaction sequence number for the GTID where these were enabled.

Now, when a temporary table is created in a session, the binary logging format is tracked. The `DROP TEMPORARY TABLE IF EXISTS` statement is only logged at the end of the session if statement-based format was in effect when the temporary table was created, so the `CREATE TEMPORARY TABLE` statement was logged. If row-based or mixed-format binary logging was in use when the table was created, the `DROP TEMPORARY TABLE IF EXISTS` statement is not logged.

Thanks to Laurynas Biveinis for the patch. (Bug #24670909, Bug #83003, Bug #28606948)

• Concurrent `INSERT`, `ALTER TABLE`, and `DROP DATABASE` operations could result in deadlock. (Bug #24510948, Bug #82704)

• Under some conditions, the `audit_log` plugin could recursively lock a mutex, resulting in an unresponsive server. (Bug #24437533)

• In some cases, the optimizer chose a Loose Index Scan (QUICK_GROUP_MIN_MAX_SELECT) for a `GROUP BY` query even when there was a predicate with a disjunction. This is fixed by not performing a range scan when the condition in the `WHERE` clause results in more than one disjoint range tree. (Bug #24423143)

• Incorrect results could occur when the optimizer chose an index on a generated column to fetch values. (Bug #24345509)

• Debug symbol packages are now included for all `apt` platforms (previously, they were only available on Debian 9). (Bug #24008883, Bug #27990381)

• With `SQL_MODE=''`, `UNIX_TIMESTAMP(COUNT(1))` returned `NULL` instead of 0 as expected. (Bug #23529242)

• When deleting rows from a table that had an indexed virtual `BLOB` column with a NOT NULL constraint, and the generated column expression evaluated to `NULL` in one of the rows that were being deleted, conversion of the `NULL` to its NOT NULL equivalent was not performed correctly. (Bug #23321196)

References: See also: Bug #23037025, Bug #21345972.

• Setting the `MYSQL_GROUP_SUFIX` environment variable had no effect. (Bug #23072792)

• A missing check for error handling during generated column evaluation could result in a server exit. (Bug #23021693)

• To check whether a table was empty, `ALTER TABLE` performed a table scan, which is inefficient. (Bug #22688065)

• Failure to acquire tablespace metadata locks for `ALTER TABLE` when a `LOCK TABLES` was active could cause an assertion to be raised. (Bug #22486020, Bug #79820)

• Constant propagation is no longer performed when a constant expression contains a reference to the column it is meant to replace. (Bug #20964700)
• Queries with many left joins were slow if join buffering was used (for example, using the block nested loop algorithm). (Bug #18898433, Bug #72854)

• REGEXP failed to find matches occurring after a \0 character in the string expression. (Bug #17541193, Bug #70470)

• Selecting from a view that involved aggregation and WITH ROLLUP could result in a spurious Column col_name cannot be null error. (Bug #11755860, Bug #47693)

• In some cases, the row estimate used by the server to determine whether to use sampling could be inaccurate. This was because the histogram process assumed that the estimate for the number of rows in the table was current, although it was not updated by (for example) INSERT or DELETE statements. Now the histogram process requests an updated count of rows. (Bug #88710, Bug #27197709)

• Long-running regular expression matches could not be killed. (Bug #88676, Bug #27183583)

• An optimizer SET_VAR hint (see Variable-Setting Hint Syntax) setting cte_max_recursion_depth was ignored. (Bug #88594, Bug #27153338)

• When handling range frames, if the first row for a range frame was found, its position was not stored. This could later cause retrieval of the row from the frame buffer to fail. (Bug #88568, Bug #27149369)

• The server did not handle triggers activated by LOAD DATA correctly when --skip-log-bin was enabled. (Bug #88516, Bug #27128534)

References: This issue is a regression of: Bug #27041382.

• Stored procedures performing XA transactions and acting on views were not executed correctly. (Bug #88326, Bug #27058931)

• When a table contained a column whose length was zero, the optimizer could in some cases allocate a record buffer that was too small to hold the columns read by the query. (Bug #88283, Bug #27041288)

• A trigger containing invalid syntax, followed by an INSERT that did not specify a column list, attempted to insert a new row regardless. (Bug #88274, Bug #27041382)

• Window functions did not always produce correct results with LAST_VALUE() and frames having multiple ORDER BY expressions. (Bug #88186, Bug #27010574)

• A fix for a previous issue caused the aggregated data type to be set to VARCHAR whenever the result type was string and the column size was larger than 255 characters (maximum length for CHAR). This caused problems for data types such as JSON whose result type is a string, but which support field values longer than 255 characters. Now in such cases the data type is set explicitly to VARCHAR when an aggregated column is of type CHAR or BINARY (both represented internally as strings) but its size exceeds the maximum for CHAR. (Bug #88073, Bug #26960106)

References: This issue is a regression of: Bug #83895, Bug #25123839.

• A prepared statement containing an ORDER BY list that referred to a parameter was not always handled correctly. (Bug #87863, Bug #26867652)

• DENSE_RANK() did not work correctly for the first row in a partition when buffering was in use, due to premature initialization of the cache comparator for ORDER BY. (Bug #87760, Bug #26802696)

• The optimizer chose a composite index for ref access where only the first part of the key could be used. The composite key was suitable but was seen as a higher cost. This was because, when choosing between ref access and range access on the same index, we prefer range if certain criteria are fulfilled, one of these being to choose to avoid ref-access if it has an overly-optimistic or unrealistically
low cost as can happen when records_per_key is very low. This was done even if the estimate of the number of rows for range access was more reliable than the estimate for ref access. (Bug #87613, Bug #26727773)

References: See also: Bug #23259872.

• When a stored function was used with a table column value as an argument in a WHERE predicate, its internal not_null_tables property was falsely set to a nonempty value. If this predicate was applied to an outer join operation and one of the arguments was from an inner table of the outer join, the predicate was sometimes used (incorrectly) to convert the outer join to an inner join. According to the SQL standard, only functions that have the RETURNS NULL ON NULL INPUT property should behave in that manner. Since MySQL does not currently implement this property, stored functions are changed such that they no longer implement the RETURNS NULL ON NULL INPUT behavior. (Bug #86922, Bug #26389402)

• A view or derived table contained incorrect data when defined using a SELECT that performed aggregation of a column, and whose result was filtered with HAVING. (Bug #86840, Bug #26360114)

• The server handled triggers and generated columns incorrectly. (Bug #86637, Bug #26251621)

• A query that grouped results on a subquery which returned a BLOB (or a type based on BLOB such as JSON) sometimes failed to find the group boundaries, and so returned incorrect results. (Bug #78787, Bug #21974696)

Changes in MySQL 8.0.3 (2017-09-21, Release Candidate)

Note

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running mysql_upgrade. For example, you may find it necessary to dump your data with mysqldump before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)

• Account Management Notes
• Atomic DDL Notes
• C API Notes
• Character Set Support
• Compilation Notes
• Configuration Notes
• Data Dictionary Notes
• Deprecation and Removal Notes
• InnoDB Notes
• Optimizer Notes
Account Management Notes

- `CREATE USER` now permits a `DEFAULT ROLE` clause enabling the account default roles to be specified.

- `SHOW CREATE USER` now displays the account default roles if the default is not `NONE`. (Bug #24670738, Bug #82987)

- MySQL now maintains information about password history, which makes it possible to enable restrictions on reuse of previous passwords. DBAs can require that new passwords not be selected from previous passwords for some number of password changes or period of time. It is possible to establish password-reuse policy globally using the `password_history` and `password_reuse_interval` system variables, as well as on a per-account basis using the `CREATE USER` and `ALTER USER` statements. Together with existing password-expiration capabilities to require that passwords be changed periodically, the new reuse-restriction capabilities provide DBAs more complete control over password management. For more information, see Password Management.

  **Important**

  The implementation of password-reuse restrictions involves a change to the structure of the `mysql.user` system table and a new `mysql.password_history` system table. If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these system database changes. Until this is done, password changes are not possible.

Atomic DDL Notes

- MySQL now supports atomic data definition statements (atomic DDL). An atomic DDL statement combines the data dictionary updates, storage engine operations, and binary log writes associated with a DDL operation into a single, crash-safe, transaction that is either fully committed or rolled back.

  Both table and nontable DDL statements are supported. Table-related DDL operations require storage engine support, whereas nontable DDL operations do not. Currently, the InnoDB storage engine supports atomic DDL.

  - Supported table DDL statements include `CREATE`, `ALTER`, and `DROP` statements for databases, tablespaces, tables, and indexes, and the `TRUNCATE TABLE` statement.
Supported nontable DDL statements include:

- `CREATE` and `DROP` statements, and, if applicable, `ALTER` statements for stored programs, triggers, views, and user-defined functions (UDFs). Atomic DDL support for `CREATE TRIGGER` and `DROP TRIGGER` was added in MySQL 8.0.0.

- Account management statements: `CREATE`, `ALTER`, `DROP`, and, if applicable, `RENAME` statements for users and roles, as well as `GRANT` and `REVOKE` statements. Atomic DDL support for account management statements was added in MySQL 8.0.1.

For table-related DDL operations, InnoDB writes DDL logs to the `mysql.innodb_ddl_log` data dictionary table. Enabling the `innodb_print_ddl_logs` configuration option prints DDL recovery logs to `stderr`.

The atomic DDL feature changes the behavior of some statements:

- `DROP VIEW` fails with an error if a named view does not exist, and no changes are made. Previously, the statement returned an error indicating which views did not exist, but also dropped the views that did exist.

- `DROP TABLE` fails with an error if a named table does not exist, and no changes are made. Previously, the statement returned an error indicating which tables did not exist, but also dropped the tables that did exist.

- `DROP TABLE` is fully atomic if all named tables use an atomic DDL-supported storage engine.

- `DROP DATABASE` is atomic if all tables use an atomic DDL-supported storage engine. However, removal of the database directory from the file system occurs last and is not part of the atomic transaction. If removal of the database directory fails due to a file system error or server halt, the `DROP DATABASE` transaction is not rolled back.

- Interrupted DDL operations on tables that use an atomic DDL-supported storage engine no longer introduce discrepancies between the storage engine, data dictionary, and binary log, or leave behind orphan files.

- Partial execution of account management statements is no longer permitted. Account management statements either succeed for all named users or roll back and have no effect if an error occurs.

Changes to `DROP TABLE`, `DROP VIEW`, and account management statement behavior have implications for cross-version replication configurations.

For more information, see Atomic Data Definition Statement Support. (Bug #24620918)

**C API Notes**

- The MySQL C API now enables clients to specify that metadata transfer for result sets is optional. Suppression of metadata transfer can improve performance, particularly for sessions that execute many queries that return few rows each. For more information, see C API Optional Result Set Metadata.

**Character Set Support**

- MySQL now supports Russian collations for the `utf8mb4` Unicode character set:
  - `utf8mb4_ru_0900_ai_ci` is accent insensitive and case insensitive.
  - `utf8mb4_ru_0900_as_cs` is accent sensitive and case sensitive.
Compilation Notes

- For debug builds, the `SAFE_MUTEX` compilation flag was disabled if the `memcached` plugin was included in the build. This no longer occurs; `SAFE_MUTEX` is always enabled for debug builds. Some code issues found as a result of this change were corrected. (Bug #26442367, Bug #87068)

- Binary packages on EL6 and EL7 now are compiled using Devtoolset 6 rather than Devtoolset3 and GCC 6.2.1 rather than 4.9.2. (Bug #26436968, Bug #87061)

- MySQL now compiles for SPARC on Oracle Linux. (Bug #26306331, Bug #86745)

- MySQL compilation on macOS using Clang now requires a Clang version different from 8.0, which has problems with certain inline constructs. (Bug #26279510, Bug #86711)

- Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

Configuration Notes

- The `performance_schema_max_mutex_classes` system variable default value has been increased from 220 to 250. The `performance_schema_max_thread_classes` system variable default value has been increased from 50 to 100. (Bug #26193630)

- The new `cte_max_recursion_depth` system variable implements a common table expression (CTE) maximum recursion depth. The server terminates execution of any CTE that recurses more levels than the value of this variable. For more information, see Limiting Common Table Expression Recursion. (Bug #26136509, Bug #86444)

- The `back_log` system variable default value is now the value of `max_connections`, which enables the permitted backlog to adjust to the maximum permitted number of connections.

- To enable the Event Scheduler by default, the `event_scheduler` system variable default value was changed from `OFF` to `ON`.

- The `max_allowed_packet` system variable default value has been increased from 4194304 (4M) to 67108864 (64M).

- The `max_error_count` system variable default variable has been increased from 64 to 1024.

Data Dictionary Notes

- These `INFORMATION_SCHEMA` tables have been reimplemented as views on data dictionary tables:

  - `FILES`
  - `PARTITIONS`
  - `REFERENTIAL_CONSTRAINTS`

Queries on those tables are now more efficient because they obtain information from data dictionary tables rather than by other, slower means. For example, the server no longer must create a temporary table for each query of the `INFORMATION_SCHEMA` table.

Also as a result of this change, foreign key information is retrieved from the data dictionary, not from `InnoDB`. 
If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes. (Bug #25583288)

**Deprecation and Removal Notes**

• **Replication**: The deprecated global scope for the `sql_log_bin` system variable has been removed. `sql_log_bin` now has session scope only. Applications that rely on accessing `@@GLOBAL.sql_log_bin` should be adjusted.

• These encryption-related deprecated items have been removed:
  • The `ENCODE()` and `DECODE()` functions.
  • The `ENCRYPT()` function.
  • The `DES_ENCRYPT()`, and `DES_DECRYPT()` functions, the `--des-key-file` option, the `have_crypt` system variable, the `DES_KEY_FILE` option for the `FLUSH` statement, and the `HAVECRYPT` CMake option.

In place of the removed encryption functions: For `ENCRYPT()`, consider using `SHA2()` instead for one-way hashing. For the others, consider using `AES_ENCRYPT()` and `AES_DECRYPT()` instead. (Bug #26493987)

• The deprecated `tx_isolation` and `tx_read_only` system variables have been removed. Use `transaction_isolation` and `transaction_read_only` instead.

• The deprecated query cache has been removed. Removal includes these items:
  • The `FLUSH QUERY CACHE` and `RESET QUERY CACHE` statements.
  • These system variables: `query_cache_limit`, `query_cache_min_res_unit`, `query_cache_size`, `query_cache_type`, `query_cache_wlock_invalidate`.
  • These status variables: `Qcache_free_blocks`, `Qcache_free_memory`, `Qcache_hits`, `Qcache_inserts`, `Qcache_lowmem_prunes`, `Qcache_not_cached`, `Qcache_queries_in_cache`, `Qcache_total_blocks`.
  • These thread states: `checking privileges on cached query`, `checking query cache for query`, `invalidating query cache entries`, `sending cached result to client`, `storing result in query cache`, `Waiting for query cache lock`.
  • The `SQL_CACHE SELECT` modifier.

These deprecated query cache items remain deprecated, but have no effect, and will be removed in a future MySQL version:

• `SQL_NO_CACHE SELECT` modifier.

• The `ndb_cache_check_time` system variable.

The `have_query_cache` system variable remains deprecated, always has a value of `NO`, and will be removed in a future MySQL version.

• The deprecated `EXTENDED` and `PARTITIONS` keywords for the `EXPLAIN` statement have been removed. These keywords are unnecessary because their effect is always enabled.
• The unused `date_format`, `datetime_format`, `time_format`, and `max_tmp_tables` system variables have been removed.

• The deprecated `multi_range_count` system variable has been removed.

• The deprecated `log_warnings` system variable and `--log-warnings` server option have been removed. Use the `log_errorVerbosity` system variable instead.

• The deprecated `secure_auth` system variable and `--secure-auth` client option have been removed. The `MYSQL_SECURE_AUTH` option for the `mysql_options()` C API function was removed.

• The deprecated `--ignore-builtin-innodb` server option and `ignore_builtin_innodb` system variable have been removed.

InnoDB Notes

• Renaming of columns in a parent foreign key is temporarily disabled due to ongoing work on foreign key locking. This restriction will be lifted in MySQL 8.0.4. (Bug #26334071)

References: See also: Bug #26659110.

Optimizer Notes

• The optimizer now supports a `SET_VAR` hint that sets the session value of a system variable for the duration of a single statement. Examples:

```sql
SELECT /*+ SET_VAR(sort_buffer_size = 16M) */ name FROM people ORDER BY name;
INSERT /*+ SET_VAR(foreign_key_checks=OFF) */ INTO t2 VALUES(2);
```

For more information, see [Optimizer Hints](https://dev.mysql.com/doc/refman/8.0/en/optimizer-hints.html). (Bug #22906815)

• The optimizer now uses column-value histogram statistics stored in the `column_statistics` data dictionary table to construct query execution plans. Histogram use applies to predicates involving comparison of a column to a constant. See [Optimizer Statistics](https://dev.mysql.com/doc/refman/8.0/en/optimizer-statistics.html).

• Previously, there was no way of skipping the use of index dives to estimate index usefulness, except by using the `eq_range_index_dive_limit` system variable. Now index dive skipping is possible for single-table queries under certain query conditions (see [Range Optimization](https://dev.mysql.com/doc/refman/8.0/en/optimizer-range.html)).

• The `optimizer_switch` system variable has a new flag named `use_invisible_indexes` to control whether the optimizer uses invisible indexes for query execution plan construction. If the flag is `off` (the default), the optimizer ignores invisible indexes (the same behavior as prior to the introduction of this flag). If the flag is `on`, invisible indexes remain invisible but the optimizer takes them into account for execution plan construction.

Packaging Notes

• `mysqlcheck` was missing in the MySQL Server Docker image, which prevented `mysql_upgrade` from running. (Bug #26400146, Bug #86968)

• For Debian, non-debug binaries were moved from the `mysql-server` package to the `mysql-server-core` package. (Bug #26382333, Bug #86899)

• The Debian/Ubuntu `mysql-community-source` package is no longer produced because the MySQL source tarball it contained is provided by other packages at `dev.mysql.com`. (Bug #26201482)
Performance Schema Notes

- As of MySQL 8.0.2, Performance Schema table definitions are maintained internally to the server. In consequence of that change, `CREATE TABLE` and `DROP TABLE` are no longer possible for Performance Schema tables. (Bug #26136994)

- The `events_statements_summary_by_digest` table now provides, for each row, a sample statement that produces the digest value in the row. Applications can use this information as a more efficient means of capturing statement samples than alternatives such as probing the `xxx_history_long` tables. The latter approach requires enabling the corresponding `xxx_history_long` consumers, which is additional overhead for applications that do not otherwise need those tables. For more information, see Performance Schema Statement Digests and Sampling, and Statement Summary Tables.

  Additionally, the `FIRST_SEEN` and `LAST_SEEN` timestamp columns of the `events_statements_summary_by_digest` table now have a fractional seconds part.

- The Performance Schema `setup_instruments` table now has columns for instrument metadata: Instrument properties, instrument volatility, and a documentation string describing the instrument purpose. Also, the `TIMED` column now can be `NULL`, indicating that the instrument does not support timing. See The `setup_instruments` Table.

  The new Performance Schema `setup_threads` table exposes instrumented thread class names and attributes. See The `setup_threads` Table.

Plugin Notes

- The new `get_sysvar_source` plugin service enables plugins to retrieve the source of system variable settings.

  **Note**
  
  As part of this work, the compound-part structured-variable syntax used for referring to multiple MyISAM key caches is deprecated. See Multiple Key Caches.

Security Notes

- A new `caching_sha2_password` authentication plugin is available. Like the `sha256_password` plugin, `caching_sha2_password` implements SHA-256 password hashing, but uses caching to address latency issues at connect time. It also supports more connection protocols and does not require linking against OpenSSL for RSA password-exchange capabilities. See Caching SHA-2 Pluggable Authentication.

  - If the server is started with the `--skip-grant-tables` option to disable authentication checks, the server now enables `--skip-networking` to prevent remote connections.

Server Administration

- MySQL now supports creation and management of resource groups, and permits assigning threads running within the server to particular groups so that threads execute according to the resources available to the group. Group attributes enable control over its resources, to enable or restrict resource consumption by threads in the group. DBAs can modify these attributes as appropriate for different workloads. Currently, CPU time is a manageable resource, represented by the concept of “virtual CPU” as a term that includes CPU cores, hyperthreads, hardware threads, and so forth. The server determines at startup how many virtual CPUs are available, and database administrators with appropriate privileges
can associate these CPUs with resource groups and assign threads to groups. For more information, see Resource Groups.

Spatial Data Support

- **Incompatible Change:** Previously, these functions that test geometry relationships supported only Cartesian spatial reference systems (SRSs): `ST_Contains()`, `ST_Crosses()`, `ST_Disjoint()`, `ST_Equals()`, `ST_Intersection()`, `ST_Overlaps()`, `ST_Touches()`, `ST_Within()`, `MBRContains()`, `MBRCoveredBy()`, `MBRCovers()`, `MBRDisjoint()`, `MBREquals()`, `MBRIntersects()`, `MBROverlaps()`, `MBRTouches()`, `MBRWithin()`.

These functions now detect geometry arguments in a geographic (ellipsoidal) SRS and return geographic results. Calculations for projected SRSs and SRID 0 remain the same. For more information, see Spatial Relation Functions That Use Object Shapes, and Spatial Relation Functions That Use Minimum Bounding Rectangles.

Note

If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

- **InnoDB:** Spatial reference identifier (SRID) support was added for InnoDB spatial indexes.

- Spatial data types now permit an SRID attribute, to explicitly indicate the spatial reference system (SRS) for values stored in the column. See Spatial Data Types.

To indicate each column's SRID attribute value, if there is one, the INFORMATION_SCHEMA.COLUMNS table now has an SRS_ID column.

A spatial column with an explicit SRID attribute is SRID-restricted: The column takes only values with that ID, and SPATIAL indexes on the column become subject to use by the optimizer. The optimizer ignores SPATIAL indexes on spatial columns with no SRID attribute. See SPATIAL Index Optimization.

SQL Syntax Notes

- **ALTER TABLE** now supports easier column renaming using RENAME COLUMN old_name TO new_name syntax. See ALTER TABLE Syntax. For changing a column name but not its definition, RENAME COLUMN is more convenient than CHANGE, which requires respecifying the current column definition. With CHANGE, you must look up the definition if you do not know it, and if you do not respecify it exactly, there is a possibility of data change or loss. (Bug #11746522, Bug #26949, Bug #11747473, Bug #32497, Bug #11765084, Bug #58006, Bug #14031617)

X Plugin Notes

- The X Plugin could not be installed when the server was started with the --skip-grant-tables option. (Bug #26516678)

- X Plugin socket connections were not working correctly. (Bug #26427112, Bug #87019)

- When compiling MySQL from source, certain infrequently used CMake arguments caused issues for the X Plugin build. (Bug #26141933)

- A Mysqlx.Connection.CapabilitiesGet request using MySQL X Protocol did not return the complete list of available authentication mechanisms. (Bug #26044113)
• For mixed case or uppercase schema names, the statement `list_objects` could incorrectly report a collection as a table. (Bug #25769683)

• The X Plugin was omitted from the list of plugins to include for testing data directory permissions. (Bug #24823999)

Functionality Added or Changed

• **InnoDB**: The new `innodb_dedicated_server` configuration option, which is disabled by default, can be used to have InnoDB automatically configure the following options according to the amount of memory detected on the server:

  • `innodb_buffer_pool_size`
  • `innodb_log_file_size`
  • `innodb_flush_method`

  This option is intended for MySQL server instances that run on a dedicated server. For more information, see [Enabling Automatic Configuration for a Dedicated MySQL Server](#).

• **InnoDB**: Renaming a general tablespace is now supported by `ALTER TABLESPACE ... RENAME TO` syntax.

  The `ALTER TABLESPACE` and `DROP TABLESPACE ENGINE` clause is deprecated and will be removed in a future MySQL version.

• **InnoDB**: Code related to obsoleted InnoDB system tables was removed. `INFORMATION_SCHEMA` views based on InnoDB system tables were replaced by internal system views on data dictionary tables. Affected InnoDB `INFORMATION_SCHEMA` views were renamed:

Table 1 Renamed InnoDB Information Schema views. The first column shows the old name. The second column shows the new name.

<table>
<thead>
<tr>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>INNODB_SYS_COLUMNS</code></td>
<td><code>INNODB_COLUMNS</code></td>
</tr>
<tr>
<td><code>INNODB_SYS_DATAFILES</code></td>
<td><code>INNODB_DATAFILES</code></td>
</tr>
<tr>
<td><code>INNODB_SYS_FIELDS</code></td>
<td><code>INNODB_FIELDS</code></td>
</tr>
<tr>
<td><code>INNODB_SYS_FOREIGN</code></td>
<td><code>INNODB_FOREIGN</code></td>
</tr>
<tr>
<td><code>INNODB_SYS_FOREIGN_COLS</code></td>
<td><code>INNODB_FOREIGN_COLS</code></td>
</tr>
<tr>
<td><code>INNODB_SYS_INDEXES</code></td>
<td><code>INNODB_INDEXES</code></td>
</tr>
<tr>
<td><code>INNODB_SYS_TABLES</code></td>
<td><code>INNODB_TABLES</code></td>
</tr>
<tr>
<td><code>INNODB_SYS_TABLESPACES</code></td>
<td><code>INNODB_TABLESPACES</code></td>
</tr>
<tr>
<td><code>INNODB_SYS_TABLESTATS</code></td>
<td><code>INNODB_TABLESTATS</code></td>
</tr>
<tr>
<td><code>INNODB_SYSVIRTUAL</code></td>
<td><code>INNODB_VIRTUAL</code></td>
</tr>
</tbody>
</table>

After upgrading to MySQL 8.0.3 or later, update any scripts that reference previous InnoDB `INFORMATION_SCHEMA` view names.

The new `INFORMATION_SCHEMA.INNODB_TABLESPACES_BRIEF` view provides space, name, path, flag, and space type data for InnoDB tablespaces.
MySQL 8.0 Release Notes

- **InnoDB:** When InnoDB was integrated with the global data dictionary, file-per-table tablespace names in the data dictionary were created in the form of innodb_file_per_table_\textit{x}, where \textit{x} is the InnoDB tablespace ID. For ease of use, file-per-table tablespace names in the data dictionary are once again the same as the table name.

  Upgrading from MySQL 5.7 to MySQL 8.0 appends MySQL 5.7 innoDB_table_stats and innoDB_index_stats tablespace names in the data dictionary with "\_backup57" to differentiate them from their MySQL 8.0 counterparts.

- **InnoDB:** The default \texttt{innodb_autoinc_lock_mode} setting was changed from 1 (consecutive) to 2 (interleaved). Interleaved lock mode permits the execution of multi-row inserts in parallel, which improves concurrency and scalability. The new \texttt{innodb_autoinc_lock_mode} default setting reflects the change from statement-based replication to row based replication as the default replication type in MySQL 5.7. Statement-based replication requires the consecutive auto-increment lock mode to ensure that auto-increment values are assigned in a predictable and repeatable order for a given sequence of SQL statements, whereas row-based replication is not sensitive to the execution order of SQL statements. For more information, see InnoDB AUTO_INCREMENT Lock Modes.

  For systems that use statement-based replication, the new \texttt{innodb_autoinc_lock_mode} default setting may break applications that depend on sequential auto-increment values. To restore the previous default, set \texttt{innodb_autoinc_lock_mode} to 1.

- **InnoDB:** Serialized dictionary information (SDI) is now present in all InnoDB tablespace files except for temporary tablespace and undo tablespace files. SDI is serialized metadata for table and tablespace objects. The presence of SDI data provides metadata redundancy. For example, dictionary object metadata may be extracted from tablespace files if the data dictionary becomes unavailable. SDI extraction is performed using the \texttt{ibd2sdi} tool. SDI data is stored in JSON format.

  The inclusion of SDI data in tablespace files increases tablespace file size. An SDI record requires a single index page, which is 16KB in size by default. However, SDI data is compressed when it is stored to reduce the storage footprint.

- **InnoDB:** The \texttt{innodb_flush_neighbors} default value was changed from 1 to 0, which disables flushing of neighboring pages from the buffer pool. A setting of 0 is optimal for non-rotational storage (SSD) devices where seek time is not a significant factor. For systems that use rotational storage (HDD), it is recommended to change the setting back to the previous default value of 1.

- **InnoDB:** Default values for configuration options that affect buffer pool preflushing and flushing behavior were modified:

  - The \texttt{innodb_max_dirty_pages_pct_lwm} default value was changed to 10. The previous default value of 0 disables buffer pool preflushing. A value of 10 enables preflushing when the percentage of dirty pages in the buffer pool exceeds 10%. Enabling preflushing improves performance consistency.

  - The \texttt{innodb_max_dirty_pages_pct} default value was changed from 75 to 90. InnoDB attempts to flush data from the buffer pool so that the percentage of dirty pages does not exceed this value. The increased default value permits a greater percentage of dirty pages in the buffer pool.

- **InnoDB:** The minimum \texttt{innodb_undo_tablespaces} value changed from 0 to 2. In previous releases, the system tablespace is used for rollback segments if \texttt{innodb_undo_tablespaces} is set to 0. A minimum value of 2 ensures that rollback segments are created in undo tablespaces instead of the system tablespace. For more information, see Undo Tablespaces.

- **Replication; JSON:** Added the \texttt{binlog_row_value_options} system variable. Currently this variable can be unset, or set to the value \texttt{PARTIAL\_JSON}. This causes MySQL's row-based replication to use a compact binary log format for each update modifying only a small portion of a JSON document and
using any combination of \texttt{JSON\_SET()}, \texttt{JSON\_REPLACE()}, and \texttt{JSON\_REMOVE()}. The compact format includes only the modified parts of the JSON document, not the full document, in the after-image used for the update in the binary log. If the modification requires more space than the full document, or if it is not possible to generate a partial update, the full document is used instead.

See the description of the variable as well as \texttt{Partial Updates of JSON Values}, for more information.

\textbf{Replication:} The \texttt{IGNORE\_SERVER\_IDS} option of the \texttt{CHANGE MASTER TO} statement is now deprecated when using GTID-based replication (\texttt{gtid\_mode=ON}). With GTIDs, transactions that have already been applied are automatically ignored, so this function is not needed.

Before starting GTID-based replication, check for and clear all ignored server ID lists that have previously been set on the servers involved. The \texttt{SHOW\_SLAVE\_STATUS} statement, which can be issued for individual channels, displays the list of ignored server IDs if there is one. If there is no list, the \texttt{Replicate\_Ignore\_Server\_Ids} field is blank.

If \texttt{gtid\_mode=ON} is set for the server, a deprecation warning is now issued if you include the \texttt{IGNORE\_SERVER\_IDS} option in a \texttt{CHANGE MASTER TO} statement. A deprecation warning is also issued if you issue a \texttt{SET GTID\_MODE=ON} statement when any channel has existing server IDs set with \texttt{IGNORE\_SERVER\_IDS}. If you do receive the deprecation warning, you can still clear a list after \texttt{gtid\_mode=ON} is set by issuing a \texttt{CHANGE MASTER TO} statement containing the \texttt{IGNORE\_SERVER\_IDS} option with an empty list.

\textbf{Replication:} The following obsolete \texttt{mysqlbinlog} options are now deprecated and will be removed in a future MySQL version:

- \texttt{--short-form}, which could be used for testing to limit the output to statements alone.

- \texttt{--stop-never-slave-server-id}, which provided a server ID for connections using the \texttt{--stop-never} option. If you require this function, use the \texttt{--connection-server-id} option instead.

The deprecation warnings for these options are sent to standard error, rather than to standard output, so that they do not interfere with the operation of tools that use the output of \texttt{mysqlbinlog}.

\textbf{Replication:} The \texttt{log\_slave\_updates} system variable is now set to \texttt{ON} by default, so you do not need to specify \texttt{--log\_slave\_updates} explicitly when you start a replication slave.

The \texttt{log\_slave\_updates} system variable is read-only. If you need to prevent a replication slave from logging the updates performed by its SQL thread to its own binary log, specify \texttt{--skip\_log\_slave\_updates} when you start the slave, or specify \texttt{log\_slave\_updates=OFF} in the configuration file for the slave.

\textbf{Replication:} The \texttt{group\_replication\_communication\_debug\_options} variable has been added which enables you to dynamically filter out debugging and tracing messages per Group Replication component, such as GCS, XCOM, and so on.

\textbf{Replication:} The Group Replication thread states are now shown in the Performance Schema tables.

\textbf{JSON:} The \texttt{JSON\_MERGE()} function is renamed to \texttt{JSON\_MERGE\_PREERVE()}.

This release also adds the \texttt{JSON\_MERGE\_PATCH()} function, an RFC 7396 compliant version of \texttt{JSON\_MERGE\_PREERVE()}; its behavior is the same as that of \texttt{JSON\_MERGE\_PREERVE()}, with the following two exceptions:

- \texttt{JSON\_MERGE\_PATCH()} removes any member in the first object with a matching key in the second object, provided that the value associated with the key in the second object is not JSON \texttt{null}. 

MySQL 8.0 Release Notes

• If the second object has a member with a key matching a member in the first object, `JSON_MERGE_PATCH()` replaces the value in the first object with the value in the second object, whereas `JSON_MERGE_PRESERVE()` appends the second value to the first value.

This example compares the results of merging the same 3 JSON objects, each having a matching key "a", with each of these functions:

```sql
mysql> SET @x = '{ "a": 1, "b": 2 }',
-> @y = '{ "a": 3, "c": 4 }',
-> @z = '{ "a": 5, "d": 6 }';

mysql> SELECT JSON_MERGE_PATCH(@x, @y, @z) AS Patch,
-> JSON_MERGE_PRESERVE(@x, @y, @z) AS Preserve;
```

```
+------------------+
| Patch            |
+------------------+
| ["a": 5, "b": 2, "c": 4, "d": 6] |
+------------------+
| Preserve         |
| ["a": [1, 3, 5], "b": 2, "c": 4, "d": 6] |
+------------------+
```

`JSON_MERGE()` is still supported as an alias of `JSON_MERGE_PRESERVE()`, but is now deprecated and subject to removal in a future MySQL version.

See Functions That Modify JSON Values, for more information. (Bug #81283, Bug #23255346)

• InnoDB now uses Variance-Aware Transaction Scheduling (VATS) for scheduling the release of transaction locks when the system is highly loaded, which helps reduce lock sys wait mutex contention. Lock scheduling uses VATS when >= 32 threads are suspended in the lock wait queue.

For more information about VATS, see Identifying the Major Sources of Variance in Transaction Latencies: Towards More Predictable Databases.

Thanks to Jiamin Huang for the contribution. (Bug #25290971, Bug #84266)

• MySQL now extends metadata locks, as necessary, to tables that are related by a foreign key constraint. Extending metadata locks prevents conflicting DML and DDL operations from executing concurrently on related tables. This feature also enables updates to foreign key metadata when a parent table is modified. Previously, foreign key metadata, which is owned by the child table, could not be updated safely.

If a table is locked explicitly with `LOCK TABLES`, any tables related by a foreign key constraint are now opened and locked implicitly. For foreign key checks, a shared read-only lock (`LOCK TABLES READ`) is taken on related tables. For cascading updates, a shared-nothing write lock (`LOCK TABLES WRITE`) is taken on related tables that are involved in the operation.

If `LOCK TABLES` is active for a table in a foreign key relationship, `ALTER TABLE ... RENAME` is not permitted for that table. This is a temporary restriction, lifted in MySQL 8.0.4 by the patch for Bug #26647340.

• The binary log is now enabled by default at server startup. The `log_bin` system variable is set to ON by default, instead of OFF, even if the `--log-bin` option has not been specified. Binary logging is standard practice for production installations, so enabling it by default removes configuration and planning steps that were usually required.

To disable binary logging, you can specify the `--skip-log-bin` or `--disable-log-bin` option at startup.

The `server_id` system variable is now set to 1 by default, instead of 0. For servers in a replication topology, you must still change this setting to specify a unique server ID for each replication server.
Previously, the server could not start with `log_bin=ON` if no server ID was specified. Now, the server can start, but a warning message is issued if you did not set an explicit server ID. (From MySQL 8.0.4, the message is only informational.)

With binary logging enabled for a server, all statements that change data are logged to the server’s binary log, which is a sequence of files with a base name and numeric extension. By default, the server creates binary log files and an index file in the data directory. In MySQL 8.0.3, the default base name of these files is `host_name-bin`, using the name of the host machine. From MySQL 8.0.4, the default base name is `binlog` if you do not supply the `--log-bin` option, and `host_name-bin` if you supply the option with no string or an empty string.

You can choose the names and locations of the binary log files and index file by specifying the `--log-bin` and `--log-bin-index` options. You are recommended to specify a base name explicitly, so that if the host name changes, you can easily continue to use the same binary log file names. The `log_bin_basename` system variable holds the base name and any specified path for the binary log files.

The relay log and relay log index on a replication slave, whose names are specified by the `--relay-log` and `--relay-log-index` options, cannot be given the same names as the binary log and binary log index. From MySQL 8.0.3, the server issues an error message and does not start if the binary log and relay log file base names would be the same.

The server creates a new binary log file in the series each time it starts or flushes the logs. The server also creates a new binary log file automatically after the current file’s size reaches `max_binlog_size`, which defaults to the maximum permitted value of 1GB. In MySQL 8.0.3, binary log files expire by default after 30 days, and can then be automatically removed at startup or when the binary log is flushed. You can purge binary log files manually using the `PURGE BINARY LOGS` statement, or specify a different binary log expiration period using the `binlog_expire_logs_seconds` system variable.

Many other options are available to modify the behavior of binary logging. For more information, see The Binary Log and Binary Logging Options and Variables.

References: See also: Bug #26730000.

- The `expire_logs_days` system variable, which specifies the binary log expiration period in days, is now deprecated and will be removed in a future MySQL version. `expire_logs_days` does not provide sufficient flexibility for defining the binary log expiration period.

  `binlog_expire_logs_seconds` can be used to set the binary log expiration period in seconds. In MySQL 8.0.3, the effects of the two variables are currently cumulative. For example, if `expire_logs_days` is 1 and `binlog_expire_logs_seconds` is 43200, then the binary log is purged every 1.5 days. This produces the same result as setting `binlog_expire_logs_seconds` to 129600 and `expire_logs_days` to 0. Note that the default `expire_logs_days` setting of 30 days is currently added to the binary log expiration period if `expire_logs_days` is not specified. To use `binlog_expire_logs_seconds` alone, set `expire_logs_days`=0 explicitly.

  To disable automatic purging of the binary log, you must set both `expire_logs_days` and `binlog_expire_logs_seconds` explicitly to 0.

  References: See also: Bug #26483363.

- A new type of backup lock permits DML during an online backup while preventing operations that could result in an inconsistent snapshot. The new backup lock is supported by `LOCK INSTANCE FOR BACKUP` and `UNLOCK INSTANCE` syntax. The `BACKUP_ADMIN` privilege is required to use these statements.
• The zlib library version bundled with MySQL was raised from version 1.2.3 to version 1.2.11. MySQL implements compression with the help of the zlib library.

The zlib `compressBound()` function in zlib 1.2.11 returns a slightly higher estimate of the buffer size required to compress a given length of bytes than it did in zlib version 1.2.3. The `compressBound()` function is called by InnoDB functions that determine the maximum row size permitted when creating compressed InnoDB tables or inserting rows into compressed InnoDB tables. As a result, `CREATE TABLE ... ROW_FORMAT=COMPRESSED` or `INSERT` operations with row sizes very close to the maximum row size that were successful in earlier releases could now fail.

If you have compressed InnoDB tables with large rows, it is recommended that you test compressed table `CREATE TABLE` statements on a MySQL 8.0.3 test instance prior to upgrading.

• In MySQL 8.0.2, the system variables for the slave status logs, `master_info_repository` and `relay_log_info_repository`, were set to `TABLE` instead of `FILE` by default. In MySQL 8.0.3, the `FILE` setting for both these system variables is deprecated, and a warning is issued if it is used. The `FILE` setting will be removed in a future MySQL version.

The `TABLE` setting ensures that replication repository information is stored in InnoDB tables, rather than in files in the data directory. The use of tables makes replication resilient to unexpected halts.

The default names for the slave status logs when stored as files were `master.info` and `relay-log.info`. The names could be changed using the `--master-info-file` and `--relay-log-info-file` options, respectively. As InnoDB tables, the slave status logs are named `mysql.slave_master_info` and `mysql.slave_relay_log_info`.

To modify an existing replication slave that is using a `FILE` repository for the slave status logs to use `TABLE` repositories, convert the existing replication repositories dynamically by running the following commands:

```
STOP SLAVE;
SET GLOBAL master_info_repository = 'TABLE';
SET GLOBAL relay_log_info_repository = 'TABLE';
```

The master info log table `mysql.slave_master_info` should be protected because it contains the password for connecting to the master. When you back up the replication slave's data, ensure that you back up the `mysql.slave_master_info` and `mysql.slave_relay_log_info` tables containing the slave status logs, because they are needed to resume replication after you restore the data from the slave.

Bugs Fixed

• Incompatible Change; JSON: If a JSON object contained multiple members with the same key name, MySQL kept the first member and discarded the remainder. This contradicts RFC 7159, which suggests that duplicate key names can be handled in one of the ways listed here:

  • Report an error (or otherwise fail to parse the object)
  • Report all of the name-value pairs, including duplicates
  • Report the last name-value pair only

When a JSON text is evaluated in JavaScript, the last name-value pair is kept if multiple pairs with the same name are specified. MySQL now does likewise, and implements the last of the three options just listed, as shown here:
The fix for this issue also corrects a failure in the MySQL 8.0 server to handle insertion into a JSON column of data containing JSON arrays as the values for multiple identical keys. (Bug #86620, Bug #86866, Bug #26238736, Bug #26369555)

- **Performance; JSON:** Creating a representation of a JSON string now optimizes for the most common case—that the string to be processed contains no special characters that need to be escaped—scanning for the first special character in the string, and copying each sequence of characters which do not require escaping in a single `memcpy()` call, rather than checking each character in turn to determine whether it needed to be escaped, escaping it if so, and then copying it, one by one, as was done previously.

This fix also corrects a failure to escape the control character \u001f, or unit separator character. (Bug #86898, Bug #26388690, Bug #87722, Bug #26780307)

References: See also: Bug #25977595.

- **InnoDB:** Foreign keys records were missing from the INFORMATION_SCHEMA.INNODB_SYS_FOREIGN table due to file name encoding of the internal parent and child table names, which caused the names to exceed the permitted length. (Bug #27020089)

- **InnoDB:** A long semaphore wait occurred when executing `ALTER TABLE`, `DROP TABLE`, and `DROP DATABASE` operations. (Bug #26779650)

- **InnoDB:** Invalid error handling code was removed from a function related to tablespace import. (Bug #26595476)

- **InnoDB:** File-per-table tablespaces created prior to MySQL 5.6 caused a failure during an in-place upgrade to MySQL 8.0.2. The tablespaces were not registered with the InnoDB SYS_TABLESPACES system table, as required.

Tables with decimal columns created prior to MySQL 5.5 also caused a failure during an in-place upgrade to MySQL 8.0.2, due to a precision type mismatch. (Bug #26542296, Bug #87229)

- **InnoDB:** During MySQL installation, an unnecessary warning was reported about creation of foreign key constraint system tables. (Bug #26483335)

- **InnoDB:** A query was interrupted during concurrent `ALTER TABLE` operations due to a secondary index entry count mismatch. (Bug #26381213)

- **InnoDB:** A segmentation fault occurred when attempting to open a table that was altered while strict mode was disabled to include conflicting `TABLESPACE` and `COMPRESSION` attributes. (Bug #26375851)

- **InnoDB:** A segmentation fault occurred during a DML operation that used the TempTable storage engine. (Bug #26363837)

- **InnoDB:** Implicit row format conversion during an `ALTER TABLE ... REORGANIZE PARTITION` operation raised an invalid assertion. (Bug #26326611)
• **InnoDB**: An `ALTER TABLE` operation that rebuilt an encrypted table did not set the encryption attribute properly. (Bug #26243264)

• **InnoDB**: A memory leak was encountered on Windows when using the `TempTable` storage engine for in-memory internal temporary tables. (Bug #26237680)

• **InnoDB**: Misleading errors were produced when running Valgrind tests on a server build that was not enabled for Valgrind testing. (Bug #26037206)

• **InnoDB**: Problematic code related to dropping orphan full-text search tables caused an invalid object ID assertion failure on startup. (Bug #25998362)

• **InnoDB**: A check for discarded partitions during a DML operation only checked the first partition. Failure to check for other discarded partitions caused an assertion failure. (Bug #25942592)

• **InnoDB**: Allocated memory was not initialized before it was written to a file, resulting in a Valgrind error. (Bug #25913151, Bug #85986)

• **InnoDB**: Adding a virtual column and index in the same statement caused an error. (Bug #25899959)

• **InnoDB**: Replication lag occurred on slave instances during large update operations on tables with many partitions. (Bug #25687913, Bug #85352)

• **InnoDB**: A failure occurred during an end range comparison. (Bug #25669686)

• **InnoDB**: Enabling the `innodb_buffer_pool_load_now` setting failed in read-only mode. The event that signals the buffer pool load thread was not initialized. (Bug #25586766)

• **InnoDB**: A cursor position check by a multiversion concurrency control row search function raised an assertion. (Bug #25377592)

• **InnoDB**: The wrong variable was passed to the `row_mysql_handle_errors` routine causing an assertion failure. (Bug #25183130)

• **InnoDB**: A long wait for a dictionary operation lock held by a full-text search synchronization operation caused a server exit. (Bug #24938374, Bug #26376681, Bug #26376239)

• **InnoDB**: Assertion code was modified to account for the possibility of a transaction attempting to acquire an explicit lock on a record while another transaction converts an implicit lock to an explicit lock on the same record prior to a commit operation. (Bug #24344131)

• **InnoDB**: A `FLUSH TABLES ... FOR EXPORT` operation on an encrypted or page-compressed table raised an assertion. (Bug #22916982)

• **InnoDB**: A `DROP TABLE` operation was not permitted with an `innodb_force_recovery` setting greater than 0. (Bug #22392152)

• **InnoDB**: A `CREATE TABLE ... SELECT` operation raised an assertion failure when the newly created table was dropped before the transaction was committed. (Bug #22154768)

• **InnoDB**: An in-place `ALTER TABLE` operation that rebuilt the table and added a foreign key without specifying a foreign key constraint name failed due to a duplicate constraint name. InnoDB did not account for existing foreign key constraint names. This issue was addressed in MySQL 8.0.3, when foreign key metadata was moved to the data dictionary, and foreign key constraint name generation was moved to the SQL layer. (Bug #18199504, Bug #71616)

• **InnoDB**: A misplaced function call that locks the InnoDB data dictionary during a foreign key check was removed. (Bug #12917178, Bug #62221)
• **Partitioning:** In certain cases when fetching heap records a partition ID could be set to zero. (Bug #86255, Bug #26034430)

• **Partitioning:** It was possible for a `CREATE TABLE` statement that failed to create a partitioned InnoDB table not to be rolled back correctly. This was due to an extraneous commit made while performing a check of foreign key information. Since partitioned tables do not presently support foreign keys, this check is unnecessary, and so is no longer made in such cases. (Bug #85299, Bug #25667278)

• **Partitioning:** Queries involving NULL were not always handled correctly on tables that were partitioned by LIST. (Bug #76418, Bug #20748521)

  References: See also: Bug #86255, Bug #26034430.

• **Replication:** On a multithreaded slave, it was possible for a deadlock state to occur due to the timing of updates to the record of disk space used by the relay log. The timing of the update has now been changed so that the deadlock cannot occur. (Bug #26729635)

• **Replication:** With `slave_preserve_commit_order=1` set, a deadlock could occur between a transaction holding a shared write lock on a table, and a transaction earlier in the commit order that also required a shared write lock. (Bug #26666609)

• **Replication:** The Group Replication flow control variables now correctly permit you to have some members in a group that do not affect the minimum throughput of the flow-control mechanism, effectively ignoring those members in case they become blocked. (Bug #26537497)

• **Replication:** The unused variable `opt_reckless_slave` was removed. (Bug #26500285)

• **Replication:** When the Group Replication plugin `Delayed_initialization_thread` failed to start due to unavailable resources, a locked mutex was being kept behind which would cause issues on `Delayed_initialization_thread` destructor. The fix ensures that the mutex is unlocked when the thread fails to start. (Bug #26394678)

• **Replication:** In the case of delayed initialization of the Group Replication plugin, deployed in single-primary mode, secondaries were able to get writes through an asynchronous replication channel, which is not allowed in normal initialization of the Group Replication plugin. (Bug #26314756)

• **Replication:** If the options file contained Group Replication related settings the server could stop unexpectedly on start up. (Bug #26314472)

• **Replication:** `FLUSH LOGS` attempted to send an OK message after having already sent an error response during the commit phase. Thanks to Laurynas Biveinis for the patch. (Bug #26272158, Bug #25363745, Bug #84437)

• **Replication:** With GTIDs generated for incident log events, MySQL error code 1590 (ER_SLAVE_INCIDENT) could not be skipped using the `--slave-skip-errors=1590` startup option on a replication slave. (Bug #26266758)

• **Replication:** `COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE` was set to an incorrect value when `group_replication_recovery_complete_at="transactions_certified"` on a recovering member. (Bug #26180350)

• **Replication:** The values of the `group_replication_recovery_use_ssl` and `group_replication_recovery_ssl_verify_server_cert` variables were not being updated when configured for the Group Replication recovery channel. (Bug #26142801)

• **Replication:** When replicating a partitioned table with an index, on a replication slave where `HASH_SCAN` was specified as part of the `slave_rows_search_algorithms` setting, the slave I/O thread sometimes stopped with an error `HA_ERR_KEY_NOT_FOUND`. (Bug #26137159)
• **Replication:** It was possible to set `server_uuid` to the same value as `group_replication_group_name`. Doing so could result in unexpected behavior because GTIDs are identified by a UUID. Now it is not possible to set `server_uuid` to the same value as `group_replication_group_name`. (Bug #26035931)

• **Replication:** The system variable `pseudo_slave_mode`, which is for internal server use, sometimes raised an assertion when it was changed inside a transaction. The server no longer changes this variable inside a transaction. (Bug #26034192, Bug #86250)

• **Replication:** When write sets are used for parallelization by a replication slave (as specified by the `binlog_transaction_dependency_tracking` system variable), empty transactions are now ignored, and the handling of relay log rotation has been optimized. (Bug #25982097)

• **Replication:** The Performance Schema `replication_applier_status_by_worker` table sometimes incorrectly displayed a value for APPLYING_TRANSACTION for an inactive worker, because the table was being populated before the worker thread stopped. (Bug #25896166, Bug #85951)

• **Replication:** Attempting to uninstall the plugin while `START GROUP_REPLICATION` executed could result in unexpected behavior. (Bug #25423650, Bug #91042, Bug #28088177)

• **Replication:** In case of a failure while creating multiple slave applier worker threads, some threads would be left orphaned and their resources would not be collected. Thanks to Laurynas Biveinis for his contribution to fixing this bug. (Bug #24679056, Bug #82980)

• **Replication:** The `binlog_checksum` option cannot be changed within a transaction. MySQL cannot log this statement, as would be required inside a transaction, while the requested function is being performed on the binary log. (Bug #22914463)

• **Replication:** It was possible to start the server with invalid values for the Group Replication flow control options. Now, the `--group-replication-flow-control-min-quota`, `--group-replication-flow-control-max-quota`, and `--group-replication-flow-control-min-recovery-quota` options are validated on server startup. (Bug #87206, Bug #26531899)

• **Replication:** If hostname resolution was not working for a member in Group Replication, the error returned when attempting to connect was referring to credentials. The error message has been improved to describe the problem with hostname resolution. (Bug #86858, Bug #26368004)

• **Replication:** The Group Replication plugin no longer sets `auto_increment_increment` and `auto_increment_offset` variables when single primary mode is active. (Bug #86669, Bug #26263155)

• **Replication:** Group Replication partition threads were not visible in the Performance Schema tables. (Bug #86626, Bug #26241008)

• **Replication:** `group_replication_force_members` could be used in situations where the group was working properly, in other words a majority was reachable. This incorrect use could cause instability in the group. Therefore, its use has been restricted to the scenario for which it was created, for forming a new membership from a subset of a previous group's membership when a majority of the members are unreachable. (Bug #86359, Bug #26093967)

• **Replication:** On a member which had both Group Replication and asynchronous replication running simultaneously, asynchronous replication was not respecting the restrictions required by Group Replication, such as only using InnoDB storage engine, tables requiring primary keys and so on. This could also be encountered when running `mysqlbinlog` against the member. Now, members that are running Group Replication and asynchronous replication do not allow Group Replication's requirements to be broken. (Bug #85164, Bug #25609945)

References: See also: Bug #85781, Bug #25828806.
• **Replication:** Joining a member running a lower version to a group running a higher version resulted in the members running the higher version becoming unreachable. (Bug #85026, Bug #25568493)

• **JSON:** Containers in the internal representations of JSON objects and arrays (`Json_object` and `Json_array`) have been changed to use smart pointers rather than raw pointers to `Json_dom`, so that orphaned DOM objects are now automatically destroyed. (Bug #26161264)

• **JSON:** ASCII character 31 (\u001f, the unit separator) in a string literal within a JSON document was not quoted when the JSON document was formatted as a string (for example, by `CAST('"\u001f"' AS JSON)`). (Bug #25977959)

• **JSON:** When a `path_expression` identified a nonarray value, the `JSON_INSERT()` and `JSON_ARRAY_INSERT()` functions failed to evaluate `path_expression[0]` as equal to `path_expression`. (Bug #86213, Bug #26022576)

• **JSON:** Searches with `JSON_EXTRACT()` that used wildcards took an inordinate amount of time. (Bug #84523, Bug #25418534)

  References: See also: Bug #83959, Bug #25151440.

• Under heavy load, an infinite loop occurred in Performance Schema buffer container code. (Bug #26666274)

• MySQL-specific typedefs such as `uchar` and `my_bool` were inadvertently reintroduced into the client namespace if the `mysql.h` header file was included. (Bug #26588846, Bug #26582752, Bug #87337)

• `uint8korr()` and related macros were fixed so that they explicitly do unaligned accesses, even on x86. (Bug #26568748, Bug #87298)

• The `main.mysql_upgrade_grant`, `main.roles-upgrade`, and `auth_sec.secure_file_priv_warnings`, test cases mishandled the error log. The `sys_vars.innodb_redo_log_encrypt_basic` test case output was unstable. Thanks to Laurynas Biveinis for the patches. (Bug #26562401, Bug #87279, Bug #26575150, Bug #87313, Bug #26575142, Bug #87314, Bug #26582158, Bug #87303)

• For debug builds, with `sql_buffer_result` enabled, recursive common table expressions caused a server exit. (Bug #26556025)

• Incorrect resolution of a window function as a constant function could result in a server exit. (Bug #26500442)

• For window functions, use of `GROUP BY ... WITH ROLLUP` could cause a server exit. (Bug #26497735, Bug #26497247)

• For window functions, `JSON` columns raised an assertion. (Bug #26496733)

• For window functions, a value less than the argument for a `RANGE` frame caused a server exit. (Bug #26496645)

• Compiling with `-DWITHOUT_SERVER=1` resulted in `my_symlink.c` compilation failure due to missing `#include` for `my_dir.h`. Thanks to Christian Hesse for the patch. (Bug #26495816, Bug #87137)

• Compiling with `-DWITH_SSL=system -DWITH_ZLIB=system` assumed that the system `openssl zlib` command was available, which might not be the case. Now availability of that command is checked, and if unavailable, the `zlib_decompress` utility is built. (Bug #26494495, Bug #87123)

• yaSSL could incorrectly perform TLS cipher negotiation. (Bug #26482173)

• Some `thread_stack` settings could result in a server exit. (Bug #26438067)
For window functions, a **ROW** frame accepted noninteger arguments for the row count. (Bug #26411055, Bug #86990)

**REPLACE**(UUID(),...) expressions could be cached (improperly) and return the same value for each row of a result set. (Bug #26395601)

When building MySQL within the source tree, **make install** installed some **CMake** files into the **mysql-test** directory within the tree. (Bug #26385175, Bug #86905)

The **PROCESS_ID** column in the Performance Schema **session_connect_attrs** and **session_account_connect_attrs** tables was changed from **INT** to **BIGINT UNSIGNED** to accommodate larger process ID values. Thanks to Daniël van Eeden for the patch. (Bug #26357806, Bug #86835)

The index on a generated column, whose value was generated from **JSON_EXTRACT()**, was sometimes not used by the optimizer when it should have been. (Bug #26352119)

Setting the **log_error_services** system variable to **NULL** caused a server exit. (Bug #26331795)

For **tar** file packages, some test suite shared libraries were installed in the server package rather than the test package. (Bug #26329850)

**SHOW COLUMNS** for a valid view could fail. (Bug #26322203, Bug #86778)

An operation that caused renaming or removal of histogram statistics could cause a server exit. (Bug #26303972)

For a **VARCHAR** column, sorting using an explicit collation (**ORDER BY col_name COLLATE collation_name**) was much slower than with an implicit collation (no **COLLATE** clause), even if the explicit collation was the same as the implicit collation. (Bug #26286790, Bug #86710)

**SET binlog_format = ROW** produced a syntax error because **ROW** is now a reserved word. This syntax is now recognized specially to preserve backward compatibility. (Bug #26269280)

**SET PERSIST_ONLY**, should be permitted only to users who have the **SYSTEM_VARIABLES_ADMIN** and **PERIST_RO_VARIABLES_ADMIN** privileges, but was incorrectly also permitted to users with the **SUPER** privilege. (Bug #26247864)

The **information_schema_stats** configuration option, introduced in MySQL 8.0.0, was removed and replaced by **information_schema_stats_expiry**.

**information_schema_stats_expiry** defines an expiration setting for cached **INFORMATION_SCHEMA** table statistics. For more information, see Optimizing **INFORMATION_SCHEMA** Queries.

The **TABLES_DYNAMIC** and **STATISTICS_DYNAMIC** internal system views were removed. (Bug #26203731, Bug #83957)

Source packages for Debian platforms contained prebuilt debug binaries, causing build failures on any architectures other than the one on which those binaries were built. (Bug #26186911)

The UDF registration service did not work if used during server startup. (Bug #26173244)

Disabling table instrumentation by changes to the Performance Schema **setup_objects** table could cause incorrect index names in index statistics. (Bug #26162562)

A misleading error message was returned when attempting to drop a nonexistent tablespace file. (Bug #26133507, Bug #86438)
• When running `mysqlbinlog` with the `--read-from-remote-server` option, rewrite rules specified using the `--rewrite-db` option were ignored, so data was not written to the target database. (Bug #26117735, Bug #86288)

• Timestamp data copied from the data dictionary cache during a DDL operation was converted using a `time_zone` value that was no longer valid. The resulting timestamp data was incorrect, causing an error in release builds and an assertion failure in debug builds. (Bug #26091333, Bug #86290)

• Successful data dictionary updates but failure to write the binary log event could result in an inconsistent state. (Bug #26037355)

• `mysqlbinlog` now prints the full metadata for the event type `Table_map_log_event`. (Bug #26020990)

• Some string functions in an `ALTER EVENT` statement could cause a server exit. (Bug #25942505)

• Uninstalling the `daemon_memcached` plugin caused a serious error. (Bug #25909540)

• Prepared statements that used a common table expression and many `?` parameters could be slow. (Bug #25903274, Bug #85933)

• For `UPDATE` or `DELETE` statements with an `ORDER BY ... LIMIT` clause, the optimizer sometimes failed to identify a cheaper ordering method than `filesort`. (Bug #25899921)

• The `rpl_diff.inc` test case file did not find the data difference between servers. Thanks to Yura Sorokin for the patch. (Bug #25860138, Bug #85838)

• An ngram fulltext parser search query returned incorrect results and raised an assertion. (Bug #25851975)

• The combination of an index virtual generated column, a foreign key, and a trigger could cause an assertion to be raised. (Bug #25817660, Bug #85757)

• Selecting from a view could yield different results with materialization enabled versus materialization disabled. (Bug #25782811, Bug #85622)

• After using `SET PERSIST` to set the `event_scheduler` system variable, the server failed to restart if started with the `--skip-grant-tables` option. (Bug #25776940)

• For debug builds, adding an index to a table that had a foreign key relationship could raise an assertion. (Bug #25739983)

• An assertion could be raised for `MIN() / MAX()` access to system tables. (Bug #25738624)

• A failed `DROP VIEW` could be written to the binary log. (Bug #25680097)

• `SHOW CREATE VIEW` sometimes added a database name prefix to table names that was not present in the original view definition. (Bug #25634576, Bug #85176)

• The Performance Schema `variables_info` table displayed incorrect `VARIABLE_SOURCE` and `VARIABLE_PATH` values for variables set within option files specified by `!include` or `!includedir` directives. (Bug #25563891)

• Constant string propagation could fail for UCA-based collations. (Bug #25503965, Bug #84837)

• `mysqlpump` displayed incorrect progress information about the number of tables dumped. (Bug #25432850)

• Calculations for UCA 9.0.0 collations were inefficient for tailoring rules containing contraction characters. (Bug #25426632, Bug #84577, Bug #25426632, Bug #84577)
MySQL 8.0 Release Notes

• GROUP BY DESC on DECIMAL values could incorrectly group NULL with non-NULL values. (Bug #25407964, Bug #84537)

• Some mysqldump warnings went to the standard output rather than the standard error output and consequently were written to the dump file. (Bug #25380000, Bug #82992)

• NULL values generated as a result of WITH ROLLUP were replaced with the previous row's value when executing a prepared statement on a view having the GROUP BY .. WITH ROLLUP clause. (Bug #25174118)

• A server error occurred when a full text search result exceeded the innodb_ft_result_cache_limit setting. The patch for this bug also backports a related patch (Bug #21140111). (Bug #25033538)

• A parser refactoring in MySQL 8.0.1 resulted in incorrect handling of some INSERT ... ON DUPLICATE KEY UPDATE statements. These problems have been corrected. (Bug #24716127, Bug #25526439, Bug #25071305)

• For debug builds, a CREATE TABLE statement with a VARBINARY or BINARY column having a default value in hexadecimal format caused a server exit. (Bug #24679166, Bug #83020)

• If a stored function was considered a constant by the optimizer, calling it from a subquery in a NOT IN condition in the WHERE clause could cause a server exit. (Bug #23577867)

• A mysqldump memory leak was fixed. Thanks to Yura Sorokin for the patch. (Bug #23531150, Bug #81714)

• Incorrect results or a server exit could result when a query used Batched Key Access optimization and a virtual generated column was part of the join buffer. (Bug #23169112)

• If a session rolled back to a savepoint and then was killed, the statements up to the point of the savepoint could be committed. (Bug #22350047, Bug #79596)

• MySQL accepted a reference to an alias of an aggregated expression defined in an outer query block even when the reference occurred within a GROUP BY subquery where the reference was meaningless. (Bug #21974346, Bug #78785)

• For clients that used Connector/Python and authenticated using the sha256_password plugin, the server could handle connections incorrectly. (Bug #21421642)

• Some SELECT DISTINCT queries with GROUP BY could return incorrect results. (Bug #20692219, Bug #76283)

Changes in MySQL 8.0.2 (2017-07-17, Development Milestone)

Note

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running mysql_upgrade. For example, you may find it necessary to dump your data with mysqldump before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)

• Account Management Notes
• During data directory initialization or upgrade, MySQL now creates a
  'mysql.session'@'localhost' reserved account. This account is used internally by plugins to
  access the server. It is locked so that it cannot be used for client connections. (Bug #25642343)

• These system variables now are available to define mandatory roles and to enable granted roles to be
  automatically activated at client connection time:

  • mandatory_roles takes a value listing roles the server should treat as automatically granted to all
    users.

  • activate_all_roles_on_login enables control over automatic activation of all granted roles
    when users log in to the server.

  For more information, see Using Roles.

Character Set Support

• For Unicode data that uses NO PAD collations, sorting of multibyte and variable-length values has been
  improved:
• **NO PAD** collations are those based on UCA 9.0.0 and higher, such as `utf8mb4_0900_ai_ci` or `utf8mb4_ja_0900_as_cs`.

• The performance improvement is greatest for key values with short weight strings; that is, strings where the weight strings do not fill their entire permitted length. For a `VARCHAR(10)` column that uses the `utf8mb4_0900_ai_ci` collation, values may take up to 160 bytes. The string 'a' uses only 2 bytes out of a possible 160 and is more sparse than 'abcdefg hij'. But even 'abcdefg hij' uses only 20 bytes of a possible 160 and is more sparse than a string of, for example, ten copies of U+337F SQUARE CORPORATION (㍿), whose weight string requires the entire 160 bytes available.

(Bug #25750527, Bug #85546)

• MySQL now supports a new collation, `utf8mb4_0900_as_ci`, for the `utf8mb4` Unicode character set. This collation is accent sensitive and case insensitive. It is similar to the default `utf8mb4` collation (`utf8mb4_0900_ai_ci`) except that the default collation is accent insensitive.

MySQL also now supports a new Japanese collation, `utf8mb4_ja_0900_as_cs_ks`, for the `utf8mb4` Unicode character set. This collation is like `utf8mb4_ja_0900_as_cs` in that it is accent sensitive and case sensitive, but `utf8mb4_ja_0900_as_cs_ks` is also kana sensitive and distinguishes Katakana characters from Hiragana characters. `utf8mb4_ja_0900_as_cs` treats Katakana and Hiragana characters as equal for sorting. Applications that require a Japanese collation but not kana sensitivity may use `utf8mb4_ja_0900_as_cs` for better sort performance. `utf8mb4_ja_0900_as_cs_ks` uses three weight levels for sorting; `utf8mb4_ja_0900_as_cs_ks` uses four.

`utf8mb4_ja_0900_as_cs_ks` is the first collation to use the `_ks` collation suffix. Japanese collations without this suffix are not kana sensitive.

For more information, see [Unicode Character Sets](https://dev.mysql.com/doc/refman/8.0/en/charsets-codecs.html).

• These character set changes were made to MySQL client support:
  
  • These clients use a default character set of `utf8mb4` rather than `latin1`: `mysql`, `mysql_upgrade`, `mysqladmin`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqldump`, `mysqlslap`, `mysqlshow`, and `mysqltest`. (This change does not affect character set autodetection performed by `mysql`, `mysqladmin`, `mysqlcheck`, `mysqlimport`, and `mysqlshow`, as described at [Connection Character Sets and Collations](https://dev.mysql.com/doc/refman/8.0/en/character-sets.html)).

  • Client connections created using the `libmysqlclient` library use a default character set of `utf8mb4` rather than `latin1`.

  • The `mysqltest` program supports a `--default-character-set` option for specifying the character set explicitly.

### Compilation Notes

• **Performance**: The strict aliasing optimization for GCC is no longer disabled, which results in a minor improvement for MySQL single-threaded performance.

• **Linux**: MySQL now compiles on Alpine Linux. (Bug #25945568, Bug #80322)

• **Solaris**: On Solaris, server builds now use `std::atomic`, so they must be linked against `libstatomic`. (Bug #25957991)

• MySQL now compiles using Visual Studio 2017. (Bug #25788406)

• The minimum version of the Boost library for server builds is now 1.64.0. (Bug #25772329, Bug #85593)
MySQL 8.0 Release Notes

- These changes were made with respect to client program development:
  - Client programs should only need to `#include` the `<mysql.h>` header file. In particular, `<my_config.h>` should not be needed, and is no longer installed.
  - The `my_init()` function is no longer included in the list of symbols exported from `libmysqlclient`. It need not be called explicitly by client programs because it is called implicitly by other C API initialization functions.
    (Bug #25732787)
  - For the Xcode IDE, header files are added to MySQL project sources so they can be searched. (Bug #25636986)
  - Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

Component Notes

- New services are available to enable server components to register and unregister system variables (`component_sys_variable_register`, `component_sys_variable_unregister`) and to register status variables (`status_variable_registration`).
  - Two new services are available to enable server components and plugins to register and unregister user-defined-functions (UDFs): `mysql_service_udf_registration` and `mysql_service_udf_registration_aggregate` provide registration services for scalar and aggregate UDFs, respectively. These services enable components and plugins to manage UDFs for themselves, without the need for `CREATE FUNCTION` and `DROP FUNCTION` statements.
    UDFs registered using either these services or `CREATE FUNCTION` now are listed in the new Performance Schema `user_defined_functions` table. See The `user_defined_functions` Table.
  - The `mysql_string` string-manipulation service has been converted from a plugin service to a component service: a set of string service APIs for use by server components. The password validation component has been updated to use the revised service implementation.

Configuration Notes

- Previously, `LOCAL` capability for `LOAD DATA` operations was enabled by default in some contexts: The client library in MySQL binary distributions was compiled with client-side `LOCAL` capability enabled, and the `local_infile` system variable was enabled on the server side. `LOCAL` capability is now disabled by default in all contexts. Client programs must be configured explicitly to enable `LOCAL`, and the server must be run with `local_infile` enabled. See Security Issues with LOAD DATA LOCAL.
  (Bug #24511108)
- `mysqld` startup behavior has been modified as follows. In general, these changes enable more consistent or easier server startup when not using `mysqld_safe`.
  - The server executable determines its own full path name at startup and uses the parent of the directory in which it is located as the default `basedir` value. This in turn enables the server to use that `basedir` when searching for server-related information such as the `share` directory containing error messages.
  - If error log output cannot be redirected to the `--log-error` option value, error output could be lost. This is now less likely.
• If the server is started using the --daemonize option and is not connected to a tty device, a default error logging option of --log-error="" is used in the absence of an explicit logging option, to direct error output to the default log file. Previously, an error could occur under these circumstances.

• The -D and -I options now are synonyms for --daemonize and --initialize, respectively.

(Bug #20398088, Bug #75343)

• MySQL now supports a SET PERSIST_ONLY variant of SET statement syntax, for making configuration changes at runtime that also persist across server restarts. Like SET PERSIST, SET PERSIST_ONLY writes the variable setting to an option file named mysql-auto.cnf in the data directory. However, unlike PERSIST, PERSIST_ONLY does not modify the runtime global system variable value. This makes PERSIST_ONLY suitable for configuring read-only system variables that only be set can at server startup. For more information, see Using Option Files, and SET Syntax for Variable Assignment.

Use of SET PERSIST_ONLY requires the new PERSIST_RO_VARIABLES_ADMIN privilege, in addition to the SYSTEM_VARIABLES_ADMIN privilege necessary for persisting system variables.

• The explicit_defaults_for_timestamp system variable is now enabled by default (previously disabled by default), and a warning occurs if you disable it. This means that the nonstandard (and deprecated) behaviors for default values and NULL-value handling in TIMESTAMP columns are now disabled by default.

Data Dictionary Notes

• The column_stats system table has been removed and replaced by the column_statistics data dictionary table.

• The version data dictionary table was renamed to dd_properties.

Metadata for INFORMATION_SCHEMA tables created by dynamic plugins now is recorded in the data dictionary and visible through the INFORMATION_SCHEMA TABLES table.

These tables are no longer visible in INFORMATION_SCHEMA: SHOW_STATISTICS, SHOW_STATISTICS_DYNAMIC, STATISTICS_BASE, STATISTICS_DYNAMIC, TABLES_DYNAMIC. User impact is minimal because these tables are for internal use by server components.

• The InnoDB storage engine now uses the MySQL data dictionary rather than its own storage engine-specific data dictionary. For information about the data dictionary, see MySQL Data Dictionary.

The following list briefly describes the main implications of this change:

• Upgrade and downgrade implications:

  • To upgrade from MySQL 5.7 to MySQL 8.0, you must perform the upgrade procedure described at Upgrading MySQL.

  • Downgrading from MySQL 8.0 to MySQL 5.7 is only supported using the logical downgrade method (a mysqldump downgrade). In-place downgrades are not supported.

  • Metadata updates associated with exporting and importing tablespaces using the transportable tablespace feature are now performed on global data dictionary tables instead of InnoDB data dictionary tables.

  • InnoDB in-memory metadata is now instantiated from global data dictionary objects. This metadata was previously read from InnoDB system tables.
• Table options that signify tablespace encryption and transparent page compression are now retrieved from the global data dictionary.

• Data dictionary support was added for InnoDB FULLTEXT indexes. Auxiliary index table names were changed to lowercase.

• InnoDB metadata created or modified during DDL operations is now written to the global data dictionary.

• mysql system tables and data dictionary tables are now created in a single InnoDB tablespace file named mysql.ibd in the MySQL data directory. Previously, these tables were created in individual InnoDB tablespace files in the mysql database directory. Associated changes include:
  • The InnoDB data dictionary buffer table, which stores fast changing InnoDB metadata, was renamed to innodb_dynamic_metadata and moved from the InnoDB system tablespace to the data dictionary tablespace.
  • Undo tablespace metadata now resides in the data dictionary tablespace.
  • Temporary tablespace metadata now resides in the data dictionary tablespace.
  • Table definitions for tables created in the mysql tablespace, the InnoDB system tablespace (innodb_system), or general tablespaces now include a TABLESPACE attribute. This change has the following implications:
    • You cannot use CREATE TEMPORARY TABLE ... LIKE to create an empty table based on the definition of a table that resides in any of the aforementioned tablespaces, as these tablespaces do not support temporary tables.
    • CREATE TABLE ... LIKE preserves the TABLESPACE attribute of the original table and creates a new table in the defined tablespace regardless of the innodb_file_per_table setting. This is a temporary regression.

For more information and workarounds, see CREATE TABLE ... LIKE Syntax, and CREATE TEMPORARY TABLE Syntax.

Deprecation and Removal Notes

• The mysql client by default strips comments in statements sent to the server, and this behavior is controlled using --skip-comments (strip comments), and --comments (preserve comments).

Comment stripping is now deprecated. This feature and the options to control it will be removed in a future MySQL version.

• Support for these deprecated syntax constructs for table and column references has been removed and their use now results in an error. Instances of these constructs should be changed to remove the leading period.

• .col_name
• .tbl_name
• .tbl_name.col_name

• Symbolic link support as described at Using Symbolic Links for MyISAM Tables on Unix, along with the --symbolic-links option that controls it, is now deprecated and will be removed in a future MySQL
version. In addition, the option is now disabled by default. The related `have_symlink` system variable also is deprecated and will be removed in a future MySQL version.

### Function and Operator Notes

- MySQL now supports window functions that, for each row from a query, perform a calculation using rows related to that row. These include functions such as `RANK()`, `LAG()`, and `NTILE()`. In addition, most existing aggregate functions now can be used as window functions; for example, `SUM()` and `AVG()`.

  For more information, see [Window Functions](#).

**Note**

Each of the following words now is a reserved word and cannot be used as an identifier without identifier quoting: `CUME_DIST`, `DENSE_RANK`, `FIRST_VALUE`, `GROUPS`, `LAG`, `LAST_VALUE`, `LEAD`, `NTH_VALUE`, `NTILE`, `OVER`, `PERCENT_RANK`, `RANK`, `ROW_NUMBER`, `WINDOW`.

### Logging Notes

- **Incompatible Change**: These error-logging changes have been made:

  - The server is more forgiving if it cannot find the configured error-message file (specified using the `lc_messages_dir` and `lc_messages` system variables). Previously, the server wrote a message to the error log to indicate the problem, then aborted the startup process and exited. Now the server writes a message, but continues startup and defaults to built-in English messages. This applies to messages the server writes to the error log and sends to clients. See [Setting the Error Message Language](#).

  - Error logging was rewritten to use the MySQL component architecture. Traditional error logging is implemented using built-in components, and logging using the system log is implemented as a loadable component. In addition, a loadable JSON log writer is available. To control which log components to enable, use the `log_error_services` system variable. For more information, see [The Error Log](#).

  - Incompatibility: To enable logging to the system log, you must load the `log_sink_syseventlog` log component and list it in the `log_error_services` value (see [Error Logging to the System Log](#)). This differs from MySQL 5.7 and earlier, for which logging to the system log is enabled by default on Windows, and on all platforms requires no component loading.

    A consequence of this configuration change is that the `log_syslog` system variable previously used to control logging to the system log is obsolete and changes to its value have no effect. `log_syslog` is now deprecated and will be removed in a future MySQL version.

### Optimizer Notes

- MySQL now enables management of histogram statistics for table column values:

  - The `ANALYZE TABLE` statement supports `UPDATE HISTOGRAM` and `DROP HISTOGRAM` clauses for generating and removing column histogram statistics.

  - The server stores histogram information in the `column_statistics` data dictionary table. Histograms are viewable using the `INFORMATION_SCHEMA.COLUMN_STATISTICS` table.

  - The `histogram_generation_max_mem_size` system variable controls the amount of memory available for histogram generation.
• The Performance Schema has a `memory/sql/histograms` instrument for monitoring memory allocations performed for histogram generation.

• `mysqldump` and `mysqlpump` have a `--column-statistics` option to add `ANALYZE TABLE` statements to the output to generate histogram statistics for dumped tables when the dump file is reloaded.

For more information, see `ANALYZE TABLE Syntax`, `Optimizer Statistics`, and `The INFORMATION_SCHEMA COLUMN_STATISTICS Table`.

Packaging Notes

• `mysqladmin` was added to Docker/Minimal packages because it is needed by InnoDB Cluster. (Bug #25998285)

• Debian/Ubuntu packages now support multiple MySQL instances with systemd. See `Managing MySQL Server with systemd`. (Bug #24559588, Bug #82785)

Parser Notes

• The parser rules for `ALTER TABLE` were refactored to be context independent and improve maintainability and extensibility. A resulting effect is that some previously accepted undocumented syntax variants are no longer accepted. For example, `CREATE TABLE` statements were permitted with column names qualified by the table name, or by the current database and table name. Such statements now produce an error.

Performance Schema Notes

• The Performance Schema now supports versioning, and maintains the current definitions for its tables internally. At startup, the server compares its supported Performance Schema version with the Performance Schema version stored in the data dictionary. If the versions differ, the server drops any old Performance Schema tables and recreates them using the current definitions. In consequence of this change:

• For MySQL upgrades, it is no longer necessary to run `mysql_upgrade` to incorporate changes to Performance Schema tables because they are recreated automatically as necessary at server startup.

• The `mysql_system_tables.sql` support script no longer includes SQL statements for Performance Schema table creation because these table definitions are maintained internally.

To support dynamic Performance Schema table manipulation, a new component service named `pfs_table_service` is now available.

• The Performance Schema default instrumentation settings have changed:

  • The `transaction` instrument and the `events_transactions_current` and `events_transactions_history` consumers are now enabled by default. See `Performance Schema Transaction Tables`.

  • All `memory/%` memory instruments are now enabled by default. See `Memory Summary Tables`.

  • The `wait/lock/metadata/sql/ml` metadata lock instrument is now enabled by default. See `The metadata_locks Table`.
Security Notes

• The linked OpenSSL library for the MySQL Commercial Server has been updated to version 1.0.2l. Issues fixed in the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html.

This change does not affect the Oracle-produced MySQL Community build of MySQL Server, which uses the yaSSL library instead. (Bug #26160962)

• Previously, any user could execute the `XA RECOVER` statement to discover the XID values for outstanding prepared XA transactions, possibly leading to commit or rollback of an XA transaction by a user other than the one who started it. Now `XA RECOVER` is permitted only to users who have the new `XA_RECOVER_ADMIN` privilege, which is expected to be granted only to administrative users who have need for it. This might be the case, for example, for administrators of an XA application if it has crashed and it is necessary to find outstanding transactions started by the application so they can be rolled back. This privilege requirement does not affect normal commit or rollback of an XA transaction because the user who started it knows its XID.

For MySQL upgrades, `mysql_upgrade` grants `XA_RECOVER_ADMIN` to users who have the `SUPER` privilege, unless some account is found that already has `XA_RECOVER_ADMIN`. (Bug #17188129)

Spatial Data Support

• Argument suitability checking was improved for these spatial functions that test geometry relationships: `ST_Contains()`, `ST_Crosses()`, `ST_Disjoint()`, `ST_Equals()`, `ST_Intersects()`, `ST_Overlaps()`, `ST_Touches()`, `ST_Within()`, `MBRContains()`, `MBRCoveredBy()`, `MBRCovers()`, `MBRDisjoint()`, `MBREquals()`, `MBRIntersects()`, `MBROverlaps()`, `MBRTouches()`, `MBRWithin()`. For more information, see Spatial Relation Functions That Use Object Shapes, and Spatial Relation Functions That Use Minimum Bounding Rectangles.

As a consequence of this work, performance of spatial relation functions was improved for cases where both arguments contain geometries of one dimensionality; that is, where the `GeometryCollection` arguments (or one `GeometryCollection` and another type of geometry) passed to a spatial relation function can be reduced to a `MultiPoint`, `MultiLineString`, or `MultiPolygon`. In those cases, the more complex general `GeometryCollection` can be avoided. (Bug #22285402, Bug #79406)

Test Suite Notes

• The `--do-test-list` option for `mysql-test-run.pl` accepts an argument for a file containing tests one per line. Lines in the file are now accepted in any of the following formats:

```
test_name
test_name.test
suite_name.test_name
suite_name.test_name.test
path/to/test/file
```

(Bug #25700362)

• `mysql-test-run.pl` and `mysqltest` are now more restrictive about permitted test case and result file names. Names must consist only of alphanumeric characters (A-Z, a-z, 0-9), dash (-), or underscore (_), and cannot start with dash or underscore. An error occurs for nonconforming file names. (Bug #25487471)

• These changes were made to the `--xml-report` option for `mysql-test-run.pl`:

177
• A `<failure>` tag identifies any test failing on a retry attempt.

• Separate statistics and fields are included for skipped and disabled tests.

• The XML report is created in the build directory if no absolute path is given for out-of-source builds.

• For combination runs, a field named `variation` is included in the `<testcase>` tag.

• Results for all tests belonging to a suite are aggregated within a single `<testsuite>` tag.

• Information regarding failures is reported with a brief reason in an attribute named `message`, along with other details within the tag.

(Bug #25349924)

• `mysql-test-run.pl` now supports a `--report-unstable-tests` option that has these effects:

  • Reports any test that has passed using at least one retry attempt in a separate category called “Unstable tests” in the summary.

  • If all failures are due to unstable tests, `mysql-test-run.pl` produces a warning but exits successfully.

  • Adds a new XML tag to report unstable tests, if the `--xml-report` option is also specified.

(Bug #24473420, Bug #25984429)

• `mysqltest` now supports a `replace_numeric_round` command that takes an argument value from 0 to 16 indicating the number of decimals to round numeric values to. This can be used to help prevent result content mismatch errors for tests in which slightly different results are obtained across platforms due to precision differences. Thanks to Daniel Black for the patch. (Bug #23280117, Bug #81399)

• `mysql-test-run.pl` now looks for a `testname-client.opt` file, which is analogous to the `testname-master.opt` file but for specifying test-specific client options. (Bug #17084918)

X Plugin Notes

• X Plugin now handles expired SSL certificates correctly. (Bug #25835833)

• The output of clauses such as `GROUP BY` and `HAVING` has been improved by moving the `grouping_criteria` expression to the derived query. (Bug #25549637, Bug #24497007)

• During install of the X Plugin a blank rule was being generated in the Firewall white list. (Bug #24488234)

Platform-Specific Notes

• `Linux`: The generic Linux build for MySQL 8.0 now supports Non-Uniform Memory Access (NUMA) on its host system. Systems that use the build need to have `libnuma` installed on them. See Installing MySQL on Unix/Linux Using Generic Binaries for more details. (Bug #26005558)

Functionality Added or Changed

• **Incompatible Change; InnoDB:** `ADD PARTITION`, `DROP PARTITION`, `COALESCE PARTITION`, `REORGANIZE PARTITION`, and `REBUILD PARTITION ALTER TABLE` options are now supported
by native partitioning in-place APIs and may be used with `ALGORITHM={COPY|INPLACE}` and `LOCK` clauses.

`EXCHANGE PARTITION`, which previously supported `ALGORITHM` and `LOCK` clauses, is performed by a new server layer API for compatibility with the MySQL data dictionary. Partition and table exchange is performed internally by the native partition handler.

`DROP PARTITION` with `ALGORITHM=INPLACE` deletes data stored in the partition and drops the partition. However, `DROP PARTITION` with `ALGORITHM=COPY` or `old_alter_table=ON` rebuilds the partitioned table and attempts to move data from the dropped partition to another partition with a compatible `PARTITION ... VALUES` definition. Data that cannot be moved to another partition is deleted.

- **InnoDB:** InnoDB now uses tablespace map files during recovery to identify tablespaces that require redo log application. This is a change from redo log tablespace discovery that was used previously. If tablespace map files are lost or corrupted, the `innodb_scan_directories` startup option may be used to specify tablespace directories when starting MySQL after a server outage. For more information, see [Tablespace Discovery During Crash Recovery](#). (Bug #24793413)

- **InnoDB:** In previous releases, InnoDB stores `ROW_FORMAT` and `KEY_BLOCK_SIZE` attributes specified in `CREATE TABLE` or `ALTER TABLE` statements even if those attributes are silently ignored when creating or altering a table. In such cases, `SHOW CREATE TABLE` and the `Create_options` column reported by `SHOW TABLE STATUS` show the specified `ROW_FORMAT` and `KEY_BLOCK_SIZE` attributes rather than the actual attributes of the table. InnoDB now stores and reports the actual `ROW_FORMAT` and `KEY_BLOCK_SIZE` attributes of the table.

- **Replication:** In previous versions issuing `STOP GROUP_REPLICATION` stopped the plugin but the server still accepted transactions. This meant the transactions were not transmitted to the group. To make `STOP GROUP_REPLICATION` safer, now `super_read_only` is set to `ON` immediately upon issuing `STOP GROUP_REPLICATION`, which ensures no transactions are accepted. (Bug #25495067, Bug #84795)

- **Replication:** When there was a network partition and a member was in a minority all queries to that member blocked. To improve this situation, the `group_replication_unreachable_majority_timeout` variable has been added which enables you to configure how long members in a minority wait to regain contact with a member in the majority before leaving the group. (Bug #25473794)

- **Replication:** The `group_replication_transaction_size_limit` variable was added to enable you to protect a group against large transactions causing a failure. (Bug #84785, Bug #25510757)

- **Replication:** Support for binary log files created by versions earlier than MySQL 5.0 has been removed, and binary log Version 1 and Version 3 formats are no longer supported by slaves or `mysqlbinlog`.

- **Replication:** The `replication_group_members` and `replication_group_member_stats` Performance Schema tables have been extended to provide more monitoring information for Group Replication. The `replication_group_members` table now includes information about member's roles, and the versions of MySQL running on a member. The `replication_group_member_stats` table now includes information about all members in the group, the applier and local queues, and rolled back transactions.

- **Replication:** The receiver thread has been improved to no longer block other thread's activities when waiting for disk space. This improves the monitoring of replication ensuring that it reports correctly when the receiver thread is waiting for disk space. If you are not able to free disk space to allow the receiver thread to continue its activity, it can be forcefully stopped without side effects in most cases.
- **Replication:** A new transaction length field has been added to the `Gtid_log_event` which stores the transaction length in bytes.

- **Replication:** The following variables have had their defaults changed to ensure replication is as robust and efficient as possible by default:
  - `master_info_repository=TABLE` and `relay_log_info_repository=TABLE` ensure that replication repository information is stored in InnoDB tables, making replication resilient to unexpected halts.
  - `transaction_write_set_extraction=XXHASH64` enables the optimal method for generating write set hashes.
  - `slave_rows_search_algorithms='INDEX_SCAN,HASH_SCAN'` enables the use of a hash table to avoid repeated table scans when no Primary Key or Primary Key Equivalent (non-null unique secondary index) exists for a table
  - `expire_logs_days=30` causes unused binary logs that are older than 30 days to be purged, ensuring storage space is used efficiently.

- **Replication:** The Group Replication plugin now notifies other components in the server that some relevant events have happened. Upon view changes, recovery state updates, network partitioning and primary election, the plugin informs listeners registered in the service registry and notifies them that an event has occurred. These listeners, which could even be other plugins, can then react to these events.

- **Replication:** The `group_replication_member_weight` variable has been added which enables you to control the election of new primaries in single-primary mode. In previous versions primary election was based on the member's UUID, with the lowest UUID elected as the new primary in the event of fail over. Use this variable to assign numeric weights to members to ensure that specific members are elected, for example during scheduled maintenance of the primary or to ensure certain hardware is prioritized.

- **Replication:** The following new system variables have been added to configure Group Replication:
  - `group_replication_flow_control_hold_percent`
  - `group_replication_flow_control_max_commit_quota`
  - `group_replication_flow_control_member_quota_percent`
  - `group_replication_flow_control_min_quota`
  - `group_replication_flow_control_min_recovery_quota`
  - `group_replication_flow_control_period`
  - `group_replication_flow_control_release_percent`

  These variables enable you to fine tune flow control of individual group members, adjusting the quota depending on the task the member performs in the group.

- **JSON:** Added support for ranges in the XPath expressions used with many MySQL JSON functions, including `JSON_EXTRACT()` and `JSON_REMOVE()`. Such a range is specified using the syntax `start to end`, where `start` and `end` are, respectively, the first and last indexes of a range of elements from a JSON array (always numbered starting with 0). For example, `[$[1 to 3]` includes the second, third, and fourth elements, as shown here:

```sql
mysql> SELECT JSON_EXTRACT('[1, 2, 3, 4, 5]', '[$[1 to 3]');
```
This work also provides support in such expressions for the `last` keyword, which you can use to represent the index of the last (rightmost) element in the current array, like this:

```sql
mysql> SELECT JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last]');
+--------------------------------------------+
| JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last]') |
+--------------------------------------------+
| 5                                          |
+--------------------------------------------+
1 row in set (0.00 sec)
```

Indexes relative to the end of the array are also supported, as shown here:

```sql
mysql> SELECT JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last-4 to last-2]');
+--------------------------------------------------------+
| JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last-4 to last-2]') |
+--------------------------------------------------------+
| [1, 2, 3]                                              |
+--------------------------------------------------------+
1 row in set (0.00 sec)
```

For further information and examples, see Searching and Modifying JSON Values. (Bug #79052, Bug #22285926)

**• JSON:** The optimizer now supports partial (in-place) updates of JSON documents when using `JSON_SET()`, `JSON_REPLACE()`, or `JSON_REMOVE()`. (Previously, all updates of JSON column values were done by completely removing the previous document and writing the new one in its place.) In general, this optimization can be performed when the replacement value is less than or equal in size to the column's original value, and no new elements are added to the original value. It cannot be performed for a simple replacement of the column value of the form `UPDATE tbl SET json_col = json_val`.

The JSON Data Type provides more information about the requirements for an update to be optimized in this way.

This work includes the addition of two JSON utility functions, `JSON_STORAGE_SIZE()` and `JSON_STORAGE_FREE()`, in the MySQL Server. `JSON_STORAGE_SIZE()` returns the number of bytes used to store the binary representation of a JSON document, whether the document is presented as a column value in a table, as the value of a user variable, or as a JSON literal. In the case of a `JSON` column, this is the space used to store the JSON document as it was inserted into the column, prior to any partial updates that may have been performed on it since then. `JSON_STORAGE_FREE()` shows the number of bytes in the binary representation of a JSON column value that were freed by the most recent partial update of the column. For a user variable storing a JSON document, `JSON_STORAGE_FREE()` always returns 0; it also returns 0 if the argument is a JSON literal.

Each of these functions, like many other MySQL functions that act on JSON values, also accepts a string that can be successfully parsed as a JSON document. For more information and examples, see JSON Utility Functions.

**• RPM `.spec` files now include support for running unit tests. (Bug #25814143, Bug #85743)**
MySQL 8.0 Release Notes

- For Windows, MSI installer packages now include a check for the required Visual Studio redistributable package, and produce a message asking the user to install it if it is missing. (Bug #25658832)

- The `mysql` client now supports a `--binary-as-hex` option that causes display of binary data using hexadecimal notation (`0xvalue`). Thanks to Daniël van Eeden for the patch. (Bug #25340722, Bug #84391)

- The `SHOW TABLES` statement now supports an optional `EXTENDED` keyword that causes statement output to list hidden tables created by failed `ALTER TABLE` statements. (These temporary tables have names beginning with `#sql`.) For more information, see `SHOW TABLES Syntax`. (Bug #24786075, Bug #83241)

- Serialized dictionary information (SDI) files now have a lowercase extension (`.sdi`) instead of uppercase.

- The `TempTable` storage engine replaces the `MEMORY` storage engine as the default engine for in-memory internal temporary tables. The `TempTable` storage engine provides efficient storage for `VARCHAR` and `VARBINARY` columns. The `internal_tmp_mem_storage_engine` session variable defines the storage engine for in-memory internal temporary tables. Permitted values are `TempTable` (the default) and `MEMORY`. The `temptable_max_ram` configuration option defines the maximum amount of memory that can be occupied by the `TempTable` storage engine before it starts storing data on disk. The default value is 1073741824 bytes (1GiB).

  The `memory/temptable/physical_ram` and `memory/temptable/physical_disk` Performance Schema instruments may be used to monitor `TempTable` memory allocation and disk storage.

- The MySQL 8.0.2 release introduces a number of undo related changes:
  - The number of undo tablespaces can now be modified at runtime, or when the server is restarted, using the `innodb_undo_tablespaces` configuration option. Previously, the number of undo tablespaces could only be configured when initializing the MySQL instance and could not be changed afterward. This change permits the addition of undo tablespaces and rollback segments as the database grows.

    `innodb_undo_log_truncate` is enabled by default. When enabled, any undo tablespace that exceeds the threshold value defined by `innodb_max_undo_log_size` is marked for truncation. See Truncating Undo Tablespaces.

    The `innodb_undo_tablespaces` default value was changed from 0 to 2, which means that rollback segments are created in two separate undo tablespaces instead of the `InnoDB` system tablespace by default. A minimum of two undo tablespaces is required to permit truncation of undo logs.

    Setting `innodb_undo_tablespaces` to 0 is deprecated and will not be supported in a future MySQL version.

    The naming convention used for undo tablespace files is changed from `undoNNN` to `undo_NNN`, where `NNN` is the undo space number.

    The `innodb_rollback_segments` configuration option defines the number of rollback segments per undo tablespace. Previously, `innodb_rollback_segments` was a global setting that specified the total number of rollback segments for the MySQL instance. This change increases the number of rollback segments available for concurrent transactions. More rollback segments increases the likelihood that concurrent transactions use separate rollback segments for undo logs, resulting in less resource contention.

    The `innodb_undo_logs` configuration option is removed. The `innodb_rollback_segments` configuration option performs the same function and should be used instead.
MySQL 8.0 Release Notes

- The `Innodb_available_undo_logs` status variable is removed. The number of available rollback segments per tablespace may be retrieved using `SHOW VARIABLES LIKE 'innodb_rollback_segments';`

- The `SHOW COLUMNS` and `SHOW INDEX` statements now support an optional `EXTENDED` keyword that causes statement output to include information about hidden columns and indexes that MySQL uses internally and are not accessible by users. For more information, see `SHOW COLUMNS Syntax`, and `SHOW INDEX Syntax`.

Bugs Fixed

- **Incompatible Change:** Plugins such as Group Replication and X Plugin now use the `mysql.session` account added in this version. If you are upgrading from a previous version which did not include the `mysql.session` account you must run `mysql_upgrade` to ensure the account is created. If `mysql_upgrade` is not run, plugins fail to start with the error message `There was an error when trying to access the server with user: mysql.session@localhost. Make sure the user is present in the server and that mysql_upgrade was run after a server update. Previously used users, such as `mysqlxsys`, are no longer created. (Bug #26042764)

  References: See also: Bug #24311527, Bug #25642343, Bug #25750822, Bug #25103980, Bug #83841.

- **InnoDB:** Queries run on `INFORMATION_SCHEMA.TABLES` performed more slowly with `information_schema_stats set to latest`. (Bug #26197113)

- **InnoDB:** An `ALTER TABLE ... ADD PARTITION` operation that specified a `DATA DIRECTORY` clause failed to ignore the `TABLESPACE` attribute of the table. (Bug #26113652)

- **InnoDB:** When `foreign_key_checks` is disabled, a child table with a foreign key constraint can be created before the parent table, which can result in a foreign key constraint failure, as the parent table is unaware of the constraint. When a table is created, there is now a call to load foreign key constraints for the table and check for child tables. (Bug #25976199)

- **InnoDB:** A parsing error occurred while optimizing a table with a full-text index. (Bug #25914332)

- **InnoDB:** Compiling MySQL on Windows with Microsoft Visual C++ 2015 or macOS 10.12.4 with GCC 4.2.1 or Apple LLVM version 8.0.0 (clang-800.0.38) returned warnings. (Bug #25910531)

- **InnoDB:** In debug builds, an assertion was raised during bootstrap when the system tablespace file (`ibdata1`) ran out of space during creation of doublewrite pages. (Bug #25872368)

- **InnoDB:** Incorrect locking order caused a deadlock when `InnoDB` attempted to persist an auto-increment counter value to disk. (Bug #25833228)

- **InnoDB:** Internal methods for accessing dictionary table object data did not account for virtual columns. (Bug #25822154)

  References: This issue is a regression of: Bug #23748128.

- **InnoDB:** The length of a virtual column field in a virtual index record was less than the expected template column length. (Bug #25793677)

- **InnoDB:** In debug builds, shutting down the server with `--innodb-fast-shutdown=0` raised an assertion. (Bug #25756224)

- **InnoDB:** The `ibd2sdi` utility exited when run on an unsupported file type. (Bug #25738491)

- **InnoDB:** `InnoDB` did not set the compression algorithm when opening a partitioned table. (Bug #25685868)
MySQL 8.0 Release Notes

- **InnoDB**: An in-place `ALTER TABLE` operation failed to set the encryption type, causing a `FLUSH TABLES ... FOR EXPORT` operation to assert. (Bug #25672779)

- **InnoDB**: A latch that was held while registering a file close caused a hang condition. (Bug #25658467)

- **InnoDB**: During recovery, prepared transactions were rolled back if the `innodb_force_recovery` setting was greater than 0. (Bug #25651042)

- **InnoDB**: A `CREATE TABLE` operation that defined a unique key with an eight character prefix on a NOT NULL `TEXT` field would raise an assertion if a primary key was not defined. (Bug #25647413)

- **InnoDB**: Updates to data dictionary tables combined with updates to InnoDB system tables for full-text search auxiliary tables raised a lock-related assertion. (Bug #25610353)

- **InnoDB**: The server allocated memory unnecessarily for an operation that rebuilt the table. (Bug #25573565, Bug #85043)

- **InnoDB**: Test-related code intended to simulate a random read on a nonexistent page raised an invalid assertion. (Bug #25479538)

  References: This issue is a regression of: Bug #25053705.

- **InnoDB**: With `foreign_key_checks` disabled, InnoDB incorrectly printed messages to the error log when operations were performed on a table that referenced a nonexistent foreign key. (Bug #25365223)

- **InnoDB**: A `DROP TABLE` or `TRUNCATE TABLE` operation could raise an assertion following a failed online `ALTER TABLE .. ADD INDEX` operation. The index root page was dropped prematurely. (Bug #25357789)

- **InnoDB**: During a `TRUNCATE TABLE` operation on a file-per-table tablespace, a dictionary operation lock was released before eviction of dirty pages from the buffer pool, causing a latch order violation. (Bug #25357789)

- **InnoDB**: When using an index merge optimizer switch, a `SELECT COUNT(*)` operation sometimes returned 0. Partitioning code incorrectly performed a `memcpy` instead of a column copy of columns read by the index, causing the wrong records to be copied. (Bug #25332330, Bug #81031)

- **InnoDB**: After a `RENAME TABLE` operation that moved a table to a different schema, InnoDB returned an error on restart indicating that it could not locate the tablespace data file. InnoDB failed to update `INNODB_SYS_DATAFILES` data dictionary table during the `RENAME TABLE` operation. (Bug #25189192, Bug #84038)

- **InnoDB**: During an `ALTER TABLE` operation that rebuilt a table containing a virtual column, InnoDB failed to apply a concurrent insert log record. (Bug #24961167)

- **InnoDB**: InnoDB failed to apply the concurrent delete log for an in-place `ALTER TABLE` operation due to a virtual column validation issue. (Bug #24960450)

- **Partitioning**: Following execution of an `ALTER TABLE ... OPTIMIZE PARTITION` statement, names of tablespaces used by the table were not preserved, and subsequent accesses to the table eventually caused the server to fail. (Bug #25512556)

- **Replication**: When replicating a partitioned table with an index, on a replication slave where `HASH_SCAN` was specified as part of the `slave_rows_search_algorithms` setting, the slave I/O thread sometimes stopped with an error `HA_ERR_KEY_NOT_FOUND`. (Bug #26137159)

- **Replication**: A `USE` statement that followed a `SET GTID_NEXT` statement sometimes had no effect. (Bug #26128931)
MySQL 8.0 Release Notes

- **Replication:** A misleading warning was issued when the command `FLUSH LOGS` or `PURGE LOGS BEFORE` was used on a binary log file with an expiry time set, and the binary log file was in use. The warning related to the file being in use, and implied that a purge attempt had taken place, even if the expiry time had not yet been reached. Now, MySQL checks the expiry time of the binary log file first, and only then checks whether the file is in use. So the warning is only issued for an in-use binary log file that is old enough to be purged. (Bug #25973525)

- **Replication:** Groups can now contain members running different server versions to enable you to do online upgrades of a replication group. The rules for combining members in a group with different versions are:
  - If you have a group with 8.0 members, you cannot add a 5.7 member
  - If you have a group with 5.7 members you can add a 8.0 member, but it remains in read-only mode. Writing to this member is dangerous while the group contains multiple server versions and should be avoided.

  In a single-primary group, if the current primary leaves the group and a new primary must be elected, the primary is first chosen from the lower version members. If no lower version member is found, the primary is chosen from newer version members. (Bug #25876807)

- **Replication:** When `binlog_checksum=NONE` was set on a MySQL server after startup, and then Group Replication was started, if an error occurred, the server remained in `RECOVERING` state and could not be shut down. (Bug #25793366, Bug #85667)

- **Replication:** Now that XA transactions are prepared and committed in two parts, an issue with statement-based replication has been identified. If two XA transactions committed on the master are being prepared on the slave in the inverse order, locking dependencies can occur that cannot be safely resolved. The issue is not present with row-based replication.

  XA transactions are therefore now considered unsafe for statement-based replication.

  - When `binlog_format = STATEMENT`, a warning is issued for DML statements inside XA transactions, and replication might fail with deadlock on slaves.
  - When `binlog_format = MIXED`, DML statements inside XA transactions are logged using row-based replication.
  - When `binlog_format = ROW`, DML statements inside XA transactions are logged as before.

  (Bug #25786490, Bug #85639)

- **Replication:** The following Performance Schema replication tables now continue to be populated when the Performance Schema is disabled:
  - `replication_connection_configuration`
  - `replication_connection_status`
  - `replication_applier_configuration`
  - `replication_applier_status`
  - `replication_applier_status_by_coordinator`
  - `replication_applier_status_by_worker`
The exception is local timing information (start and end timestamps for transactions) in the replication tables `replication_connection_status`, `replication_applier_status_by_coordinator`, and `replication_applier_status_by_worker`. This information is not collected when the Performance Schema is disabled. (Bug #25694813)

- **Replication:** In a Group Replication setup where circular asynchronous replication was implemented between members of different replication groups, view change log events were repeatedly replicated between the groups with new generated GTIDs each time. The fix ensures that view change log events are ignored outside the named replication group where they occur, and never generate new GTIDs. (Bug #25674926)

  References: See also: Bug #26049695, Bug #25928854, Bug #25721175.

- **Replication:** When first starting the MySQL server following an installation from RPM, password validation plugin is activated by default (true only for RPM installations). If binary logging was already enabled at this time, the activation was logged, even though plugin activations should not be recorded in the binary log. (Bug #25672750)

- **Replication:** The `XA START`, `XA END`, `XA COMMIT`, and `XA ROLLBACK` statements, used to perform XA transactions, were incrementing the counter for a `replicate-do-db` filter that named the relevant database. The counter is no longer incremented for these statements. (Bug #25574200)

  References: See also: Bug #86325, Bug #26078602.

- **Replication:** Group Replication failed to start if the `super_read_only` system variable was enabled. (Bug #25481287, Bug #84733)

- **Replication:** MySQL 8.0 servers were not able to join a replication group that contained MySQL 5.7 servers, and MySQL 5.7 servers in a replication group could not be upgraded to MySQL 8.0. (Bug #25477979)

- **Replication:** In the event that a member failed to join a group the member was not stopping and continued to accept transactions. To avoid this set your members to have `super_read_only=1` in the `my.cfg` file. Group Replication now checks for this setting upon successful start up and sets `super_read_only=0`. This ensures that members which do not successfully join a group cannot accept transactions. (Bug #25474736, Bug #84728)

- **Replication:** When `mysqlbinlog` output from a MySQL 5.7 server was applied on a MySQL 8.0 server, the MySQL 8.0 server generated its own timestamp for the `original_commit_timestamp`. The fix ensures that the value of `original_commit_timestamp` is correctly set to 0, meaning that the timestamp is not known. (Bug #25316086)

- **Replication:** If the server was started with the `--log-bin` option, discovery of one Performance Schema table with invalid structure caused all subsequently checked Performance Schema tables to be marked invalid as well. (Bug #25041396)

- **Replication:** Interleaved transactions could sometimes deadlock the slave applier when the transaction isolation level was set to `REPEATABLE-READ`. (Bug #25040331)
• **Replication:** The `slave_skip_errors` system variable did not permit error numbers larger than 3000. Thanks to Tsubasa Tanaka for the patch. (Bug #24748639, Bug #83184)

• **Replication:** The SQL state of `ER_TRANSACTION_ROLLBACK_DURING_COMMIT` was HY00, which is generic and not suitable for rollback. The SQL state has been modified to 40000. (Bug #24658431)

• **Replication:** `mysqlbinlog`, if invoked with the `--raw` option, does not flush the output file until the process terminates. But if also invoked with the `--stop-never` option, the process never terminates, thus nothing is ever written to the output file. Now the output is flushed after each event. (Bug #24609402)

• **Replication:** A memory leak in `mysqlbinlog` was fixed. The leak happened when processing fake rotate events, or when using `--raw` and the destination log file could not be created. The leak only occurred when processing events from a remote server. Thanks to Laurynas Biveinis for his contribution to fixing this bug. (Bug #24323288, Bug #82283)

• **Replication:** Loading and initialization of the Group Replication plugin failed if no root account was present. (Bug #24311527)

  References: See also: Bug #25750822, Bug #25103980, Bug #83841.

• **Replication:** A slave server could lose events not yet applied when `MASTER_AUTO_POSITION=0`, both replication threads were stopped, and the applier delay was changed using `CHANGE MASTER TO MASTER_DELAY=N`. (Bug #23203678, Bug #81232)

  References: See also: Bug #25340185, Bug #84375.

• **Replication:** Transmission of large GCS messages could take so long the sender appeared to have died. (Bug #22671846)

• **Replication:** Multithreaded slaves could not be configured with small queue sizes using `slave_pending_jobs_size_max` if they ever needed to process transactions larger than that size. Any packet larger than `slave_pending_jobs_size_max` was rejected with the error `ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX`, even if the packet was smaller than the limit set by `slave_max_allowed_packet`.

  With this fix, `slave_pending_jobs_size_max` becomes a soft limit rather than a hard limit. If the size of a packet exceeds `slave_pending_jobs_size_max` but is less than `slave_max_allowed_packet`, the transaction is held until all the slave workers have empty queues, and then processed. All subsequent transactions are held until the large transaction has been completed. The queue size for slave workers can therefore be limited while still allowing occasional larger transactions. (Bug #21280753, Bug #77406)

• **Replication:** An incident event that broke replication was not written to the binary log with a GTID, so that it was not possible to skip the event using `SET gtid_next=value`. Instead, it was necessary to set the relay log file and relay log positions directly; this meant that, when autopositioning was enabled, it was necessary first to disable it, then to set the relay log file and position, and finally to re-enable autopositioning.

  Now in such cases MySQL writes the incident event into the statement cache, so that a GTID is generated and written for it prior to flushing, and that the slave applier works with the change. Then users can skip the event using the SQL statement `SET gtid_next=value`, followed by `BEGIN` and `COMMIT`. (Bug #19594845)

• **Replication:** Issuing `SHOW SLAVE STATUS FOR CHANNEL 'group_replication_recovery'` following a restart of a server using group replication led to an unplanned shutdown. (Bug #85739, Bug #25813258)
• **Replication**: Setting an empty filter rule using `CHANGE_REPLICATION_FILTER` caused issues when running tests with UBSan. (Bug #85405, Bug #25702297)

• **Replication**: When the receiver thread held a lock on the relay log while queuing an event, clients executing `SHOW SLAVE STATUS` or `SHOW RELAYLOG EVENTS` were blocked. (Bug #85084, Bug #25584734)

• **Replication**: Indexes for the Performance Schema `replication_applier_status_by_worker` table worked correctly only for single threaded replication. This was due to the fact that the indexing relied exclusively on the SQL thread ID regardless of the existence of any workers. Now in such cases, the index also considers worker thread ID when multithreading is enabled on the slave. (Bug #84646, Bug #25444610)

• **Replication**: In certain cases, the master could write to the binary log a `last_committed` value which was smaller than it should have been. This could cause the slave to execute in parallel transactions which should not have been, leading to inconsistencies or other errors. (Bug #84471, Bug #25379659)

• **Replication**: When using `group_replication_ip_whitelist=AUTOMATIC`, IPs in the private network are permitted automatically, but some class C IP addresses were not being permitted correctly. (Bug #84329, Bug #25503458)

• **Replication**: When an existing GTID_NEXT transaction was assigned a conflicting GTID by the server, Group Replication generated an assert upon detecting two transactions with same GTID. This was because Group Replication generates the GTID after conflict detection, which is later than with master/slave replication. The fix relaxes some conditions to only be called when commit is done and a message has been added to alert you when a GTID has already been used. (Bug #84153, Bug #25232042)

• **Replication**: The replication applier thread returns Error 3002 `ER_INCONSISTENT_ERROR` when there is a difference between an expected error number and the actual error number. It is now possible to ignore this error by using 3002 with `slave_skip_errors`. (Bug #83186, Bug #24753281)

• **Replication**: MySQL lost its GTID position following a restart when a dump from `mysqldump` had been used to load data.

To keep this problem from occurring, the `mysql.gtid_executed` table is now excluded automatically from dumps made by `mysqldump`. (Bug #82848, Bug #24590891)

References: See also: Bug #87455, Bug #26643180.

• **Replication**: It was possible to set `binlog_format` without causing an error when there were open temporary tables. (Bug #82467, Bug #24411680)

• **Replication**: Corruption of relay logs for one channel in multi-source replication caused good channels not to be initialized during a server restart. In addition, when run with `--skip-slave-start=false`, the server also failed to start slave threads for those channels which were in good condition, despite the fact that it should have started the slave threads for all good channels.

Now, regardless of any errors on other channels, the server attempts to create and initialize channels that are in good condition, and starts slave threads for the good channels if `--skip-slave-start` is disabled. As part of this fix, `START SLAVE` and `STOP SLAVE`, which are intended to operate on all channels, are also modified such that they continue executing on all good channels even if they find bad channels among them. (Bug #82209, Bug #24285104)

• **Replication**: The SQL thread was unable to GTID skip a partial transaction. (Bug #81119, Bug #25800025)

• **Replication**: It was possible for `FLUSH LOGS` to write a `Rotate_log_event` into an uninitialized log file. (Bug #80368, Bug #22732184)
MySQL 8.0 Release Notes

References: See also: Bug #23531998, Bug #81734.

- **Microsoft Windows:** On Windows, the Docs/INFO_SRC file was missing the **build-date** entry. (Bug #25799855)

- **JSON:** **NULLIF()** failed with an assertion error in debug builds if the result from this function was used in a JSON context. This result can now be used safely as a JSON value. This fix includes using **DEBUG_ASSERT()** for handling this error instead of **DEBUG_ABORT()**, which caused debugging problems on some platforms. (Bug #25818544)

  References: See also: Bug #21383497, Bug #21383530.

- **JSON:** The internal **Json_array** now uses a **std::vector** instead of a **Preallocated_array** to store its elements, which reduces the amount of heap space required by them. (Bug #85877, Bug #25867454)

- When the character set of one string comparison operand was a superset of the character set of the other operand, some comparisons were disallowed that should be permitted by converting the operand with the “smaller” character set to the “larger” character set. *utf8mb4* and *utf32* are considered to be a superset of any other encoding. (Bug #27897053, Bug #25642319, Bug #85224)

- The **mysql_pre_systemd** script in RPM packages found the error log setting in option files if specified as **log-error** but not as **log_error**, though both are permitted. (Bug #26148391, Bug #86466)

- **ALTER TABLE ... ADD PARTITION** statements could cause a server exit. (Bug #26132947)

- **SET PERSIST** did not work for X Plugin system variables. (Bug #26115672)

- **REFERENCES** privilege checking could use the incorrect database in some cases. (Bug #26106655)

- With the **gtid_mode** system variable not set in a **.cnf** option file or on the command-line, the server failed to restart after using **SET PERSIST** or **SET GLOBAL** to set **gtid_mode** to **ON_PERMISSIVE**. (Bug #26085712)

- During prepared statement execution, too many bytes of a buffer could be read. (Bug #26042934, Bug #85937)

- Dynamic privileges were not loaded when the server was started using the **--initialize** option. (Bug #26005645)

- Debian client packages were missing information about conflicts with akonadi-backend-mysql packages. (Bug #26002288)

- **mysqldump** could write database names in **USE** statements incorrectly. (Bug #25998635)

- If the **mysql_stmt_close()** C API function was called, it freed memory that later could be accessed if **mysql_stmt_error()**, **mysql_stmt_errno()**, or **mysql_stmt_sqlstate()** was called. To obtain error information after a call to **mysql_stmt_close()**, call **mysql_error()**, **mysql_errno()**, or **mysql_sqlstate()** instead. (Bug #25988681)

- Upgrading from MySQL 5.7 to MySQL 8.0 failed during data dictionary creation due to table names that exceeded the 64 character length limit. (Bug #25973237, Bug #86120)

- Queries could be cached incorrectly, leading to incorrect query results, under these circumstances: **InnoDB** table; rows are being inserted but have not yet been committed; a query uses the table as a base table in a derived table; the optimizer chooses to materialize the derived table. (Bug #25943038, Bug #86047)
• `mysqld` misbehaved if compiled with AddressSanitizer enabled and invoked with `--basedir=` (that is, with an empty option value). (Bug #25914296, Bug #85594)

• The `ST_LatFromGeohash()`, `ST_LongFromGeohash()`, and `ST_PointFromGeohash()` functions did not allow data that originated from a `CHAR` column. (Bug #25912557, Bug #85981)

• Attempting to drop a data file from a tablescape with multiple data files returned a duplicate file name error. (Bug #25858461)

• An attribute was added to data dictionary client method declarations to generate compilation warnings when return values are ignored. (Bug #25840927)

• The data dictionary class hierarchy was simplified. (Bug #25835968, Bug #85811)

• A restriction that prevented the data dictionary object update function (`Dictionary_client::update`) from being called twice on the same object was removed. (Bug #25833932, Bug #85800)

• On Ubuntu platforms, the MySQL service script did not terminate correctly if the user running the script did not have permission to access the data directory or PID file. (Bug #25825833)

• String comparison queries on the `INFORMATION_SCHEMA.CHARACTER_SETS.DESCRIPTION` field were case sensitive, which is a regression from earlier releases. (Bug #25824297)

• Man pages for a few utilities were missing from Debian/Ubuntu packages. (Bug #25811814)

• An `INFORMATION_SCHEMA` view executed in `LOCK TABLES` mode, using the `SERIALIZABLE` isolation level, and with `autocommit` disabled, failed to use non-locking reads, which could block DDL operations or cause deadlocks. (Bug #25811413)

• Table locking failures could occur if tables were used by a trigger for which a trigger of the same name existed in another database and the database names differed only in lettercase. (Bug #25807393)

• An in-place MySQL upgrade failed if a `.TRG` file was missing the `created` line for the trigger definition. (This problem is unlikely to occur for upgrades to MySQL 8.0 from 5.7 because the `created` line is present for MySQL 5.7.2 and higher.) (Bug #25805260, Bug #85704)

• The `field-t` unit test failed to run with AddressSanitizer enabled. Thanks to Laurynas Biveinis for the patch. (Bug #25803823, Bug #85678)

• Debian client packages were missing information about conflicts with native packages. (Bug #25799475)

• The `CREATE_OPTIONS` column of the `INFORMATION_SCHEMA.TABLES` table was treated as having the `binary` collation, so functions such as `UPPER()` and `LOWER()` did not have the intended result. (Bug #25793429)

• After importing a table with `DECIMAL` column, accessing the table raised an assertion. (Bug #25792649)

• `ALTER TABLE` could fail when the default character set changed to `utf8mb4` due to incorrect column length calculations. (Bug #25779239, Bug #85614)

• If a server component `deinit()` method failed, the component was still unloaded. (Bug #25764325)

• Stored program execution could fail for `DEFINER` accounts with expired passwords even if they were locked and could not be used to connect to the server. Now `DEFINER` accounts are not checked for expired passwords if they are locked. (Bug #25741966)

• `DISTINCT` operations on temporary tables could produce incorrect results due to allocation of too-small comparison keys. (Bug #25740550, Bug #85518)
• In optimizer trace output, `num_tmp_files` did not actually indicate number of files. It has been renamed to `num_initial_chunks_spilled_to_disk` and indicates the number of chunks before any merging has occurred. (Bug #25733784, Bug #85487)

• The Perl path in `#!` lines at the beginning of Perl scripts has been adjusted to `/usr/local/bin/perl` for FreeBSD 11. (Bug #25719975)

• With the `IGNORE_SPACE` SQL mode enabled, syntax error messages always reported line 1 as the line number. (Bug #25717617)

• `mysqldump` failed to properly quote certain identifiers in SQL statements written to the dump output. (Bug #25717383)

• Dropping a tablespace without that associated tablespace file failed with an error stating that the tablespace dictionary object is invalid. (Bug #25717019)

• Client preauthorization by the server was missing a length check for a length-encoded string. (Bug #25714674)

• In debug builds, a `CREATE TABLESPACE` operation raised an invalid assertion when using the NDB storage engine. A validation function that checked for zero-length data files did not apply to NDB tablespaces and was removed. (Bug #25700242)

• For debug builds, `EXPORT_SET()` operations could raise an assertion for some arguments. (Bug #25688192)

• For debug builds, the assertion added for Bug#59686 was too strict and could be raised when it should not have been. (Bug #25685958)

References: See also: Bug #59686.

• A code refactoring in MySQL 8.0.1 caused several assertions to be raised in debug builds. (Bug #25669590, Bug #25669606, Bug #25669580, Bug #25688504)

References: This issue is a regression of: Bug #25221172, Bug #84103.

• For some double-precision calculations, overflow could occur when calculating the exponent part. (Bug #25664323, Bug #85290)

• `SHOW INDEX` output was inconsistent for a `FULLTEXT` index defined on multiple columns. (Bug #25659276)

• During a group commit, the stack could incorrectly be reported exhausted, leading to stack overflow. (Bug #25656875)

• An in-place (binary) upgrade could change the default `character_set_client` and `character_set_connection` variables for stored routines, possibly resulting in different routine behavior. (Bug #25633041)

• For debug builds, the partitioning handler could inappropriately evaluate generated column expressions in `UPDATE` statements, resulting in a raised assertion. (Bug #25615803, Bug #85179)

• In strict SQL mode, an `ER_TRUNCATED_WRONG_VALUE` error could be converted from a warning to an error but then ignored, leading to a raised assertion. (Bug #25586959, Bug #25586673)

• Failure occurred for `ALTER TABLE` on an `ARCHIVE` table containing a `NOT NULL` column having a geometry data type. For debug builds, an assertion was raised. For non-debug builds, an error occurred. (Bug #25582178, Bug #85059)
• For debug builds, an assertion could be raised for DROP TRIGGER of a trigger for a table used by a view. (Bug #25581925)

• For upgrades from MySQL 5.7 to MySQL 8.0, the server wrote unnecessary parsing warnings to the error log. These messages are now suppressed. (Bug #25518436, Bug #84889)

• The server exited abnormally attempting to access invalid memory. (Bug #25501659)

• The maximum length of the name field in the column_type_elements table in the data dictionary was extended from 255 bytes to 1020 bytes to accommodate long ENUM and SET values that could result from using a multibyte character set.

When using a multibyte character set, a single ENUM or SET element occupies a maximum of $M \times w$ bytes in the column type definition in the data dictionary, where $M$ is the element literal length and $w$ is the number of bytes required for the maximum-length character in the character set.

In prior releases, the maximum supported length of an individual ENUM or SET element depended on the number of elements in the type. Thus, there could be a single element with $(M \times w) = 64K$, or 64K elements with $(M \times w) = 1$.

The maximum supported length of an individual ENUM or SET element is now $M \leq 255$ and $(M \times w) \leq 1020$, regardless of the number of elements in the type. (Bug #25481355)

• With mysqld secured by TCP wrappers and the hosts.allow and hosts.deny files configured to restrict access from an IP address, connection attempts from that address resulted in too many messages to the error log. (Bug #25476479, Bug #84708)

•mysqlpump no longer includes the slave_master_info and slave_relay_log_info tables in dumps of the mysql system database. Restoring a dump file containing these tables caused problems by changing the replication state improperly. (Bug #25469190)

• Changes made by calling mysql_options() to set MYSQL_OPT_SSL_MODE could be affected by later mysql_options() calls. Now setting MYSQL_OPT_SSL_MODE is unaffected by later mysql_options() calls. (Bug #25452210)

• A race condition could occur for CREATE TABLE statements with DATA DIRECTORY or INDEX DIRECTORY clauses. (Bug #25451091)

• With AddressSanitizer enabled, compiling the keyring_file plugin produced One-Definition Rule violations. (Bug #25448205)

• ALTER TABLE ... MODIFY on a DATETIME NOT NULL column using an AFTER clause resulted in an ER_INVALID_USE_OF_NULL error. (Bug #25385334)

• The range optimizer could create an incorrect query tree, resulting in a server exit. (Bug #25369742, Bug #25586531)

• mysqlld_failed to start the server if the --datadir option was specified with a relative path name. (Bug #25364806)

• XA PREPARE, XA ROLLBACK, and XA COMMIT for a transaction from a disconnected session did not take a global commit lock and modified the binary log and InnoDB redo log even when FLUSH TABLES WITH READ LOCK was in effect. This could lead to inconsistent backups when backup tools assumed that the server was in a read-only state. (Bug #25364178, Bug #84442)

• GROUP_CONCAT(DISTINCT) returned nonunique values if the data size was greater than the value of the tmp_table_size system variable. (Bug #25331425, Bug #84320)
• The fix for Bug #78777 had different effects depending on whether the Performance Schema is enabled. (Bug #25309017, Bug #84305)

References: This issue is a regression of: Bug #78777.

• An aggregate function in some nested queries could cause a server exit. (Bug #2530711)

• Virtual generated column expressions that used the `BIN_TO_UUID()`, `CONV()`, or `HEX()` functions could cause problems if the connection character set was changed. In this context, the table character set is now used for these functions regardless of connection character set. (Bug #25287633)

• The `Rewriter` plugin did not perform locking properly if the `read_only` system variable was enabled. (Bug #25264253)

• With `read_only` enabled, creation of non-TEMPORARY tables by non-SUPER users was permitted under certain conditions. (Bug #25250768)

• For a table having a `TIMESTAMP` or `DATETIME` column having a default of `CURRENT_TIMESTAMP`, the column could be initialized to `0000-00-00 00:00:00` if the table had a `BEFORE INSERT` trigger. (Bug #25209512, Bug #84077)

• On Windows, `Time` values in `SHOW PROCESSLIST` output drifted higher over time. (Bug #25101724, Bug #83019)

• An assertion was raised when a create function at server bootstrap attempted to evaluate non-resolved expressions. (Bug #24961932)

• The number of data dictionary cache lookups was reduced, and `ALTER TABLE` handling of triggers was improved help ensure trigger persistence in case of a server exit during `ALTER TABLE` processing. (Bug #24930129, Bug #83473)

• Certain queries against InnoDB tables that used a primary key and a subquery could return incorrect results if the `index_merge_intersection` flag of the `optimizer_switch` system variable was enabled. (Bug #24829050, Bug #79675)

• On x86 machines, the `uint3korr()` macro read 4 bytes of data instead of the intended 3 bytes. (Bug #24807826, Bug #83264)

• An assertion was raised during a fetch operation by the memcached plugin. (Bug #24605783)

• Queries that contained `UNION` in a subquery and `GROUP BY` could return incorrect results. (Bug #24595639)

• Some syntactically incorrect `CREATE INDEX` statements could cause a server exit rather than a syntax error. (Bug #24593992)

• An in-place MySQL upgrade failed if the `innodb_table_stats` or `innodb_index_stats` tables were not present in the `mysql` system database. (This problem should not occur for upgrades to MySQL 8.0 from 5.7 because those tables are present in MySQL 5.7.) (Bug #24557143)

• Incorrect behavior could occur for `INSERT` statements executed in stored-program or prepared-statement context, if the `VALUES` part of an `ON DUPLICATE KEY UPDATE` clause referred to a `BLOB` value in the `INSERT` column list. (Bug #24538207, Bug #25361251, Bug #25530880, Bug #25684790)

• systemd support scripts in Debian packages contained hardcoded references to the data directory, making it difficult to change the data directory using `--datadir`. (Bug #24398446, Bug #82417)

• MySQL failed to compile under macOS 10.10.5 using Clang. (Bug #24352163, Bug #82340)
• If a `REPLACE` statement tried to update a row in a table containing a virtual generated column of type `BLOB`, subsequent DML statements could behave incorrectly. (Bug #23573575)

• Some `PROXY` grants were not replicated to slaves, causing incorrect replication. (Bug #23289541, Bug #81424, Bug #23623115)

• `EXPLAIN` for single-table `UPDATE` or `DELETE` statements could raise an assertion attempting to optimize away subqueries. (Bug #23209903)

• The help output from `mysqlxtest` has been improved. (Bug #23107137, Bug #81086)

• For builds with AddressSanitizer enabled, the `ST_Simplify()` function could attempt to use already freed memory. (Bug #23023817)

• Compiler flags were adjusted to eliminate numerous warnings that occurred when compiling the `keyring_file` plugin using Clang. (Bug #22834591, Bug #80524)

• If enabling the Event Scheduler caused an event defined as `ON COMPLETION NOT PRESERVE` to be dropped because its execution time had passed, the drop event was not written to the binary log, causing slaves not to replicate it and replication failure if an event of the same name was created later. (Bug #22150112)

• For some `CREATE TABLE ... SELECT` statements, adding an `ORDER BY` clause changed column data types, or caused an assertion to be raised for debug builds. (Bug #16833464)

• `LOAD XML` performance became noticeably slower when the XML file being read contained a great many spaces, such as those introduced by indenting or pretty-printing. Now all leading whitespace is trimmed from each such value before reading it into memory. (Bug #16212207)

Changes in MySQL 8.0.1 (2017-04-10, Development Milestone)

Note
This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)

Note
This release makes several important changes in Unicode character set support. In particular, the default character set has changed from `latin1` to `utf8mb4`.

• C API Notes
• Character Set Support
• Compilation Notes
• Configuration Notes
• Data Dictionary Notes
- Deprecation and Removal Notes
- Optimizer Notes
- Packaging Notes
- Parser Notes
- Performance Schema Notes
- Plugin Notes
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
- Test Suite Notes
- X Plugin Notes
- Platform-Specific Notes
- Functionality Added or Changed
- Bugs Fixed

C API Notes

- The C API implemented by `libmysqlclient` now includes a client interface for reading a stream of replication events from a MySQL server binary log. For more information, see C API Binary Log Interface

Character Set Support

- **Important Change**: The default character set has changed from `latin1` to `utf8mb4`. These system variables are affected:
  - The default value of the `character_set_server` and `character_set_database` system variables has changed from `latin1` to `utf8mb4`.
  - The default value of the `collation_server` and `collation_database` system variables has changed from `latin1_swedish_ci` to `utf8mb4_0900_ai_ci`.

As a result, the default character set and collation for new objects differ from previously unless an explicit character set and collation are specified. This includes databases and objects within them, such as tables, views, and stored programs. One way to preserve the previous defaults is to start the server with these lines in the `my.cnf` file:

```
[mysqld]
character_set_server=latin1
collation_server=latin1_swedish_ci
```

- Performance of UCA 9.0.0-based collations (for example, `utf8mb4_0900_ai_ci`) was improved. These collations are now faster than any other UCA collations.

  Additionally, the `max_length_for_sort_data` system variable default value has been increased from 1024 to 4096. (Bug #24823885, Bug #83319)
MySQL 8.0 Release Notes

- The default collation for the utf8mb4 character set has changed from utf8mb4_general_ci to utf8mb4_0900_ai_ci; utf8mb4_general_ci does not handle characters outside the Basic Multilingual Plane (BMP) correctly. (Bug #24742157)

- The pad attribute for Unicode 9.0.0 collations was changed from PAD SPACE to NO PAD. Consequently, these collations now treat spaces at the end of strings like any other character. The affected collations have names that contain the string _0900_.

Comparisons of VARCHAR columns that have a 9.0.0 collation differ from other collations with respect to trailing spaces. For example, 'a' and 'a ' compare as different strings, not the same string. Example:

```
mysql> SET NAMES 'latin1' COLLATE 'latin1_swedish_ci';
mysql> SELECT 'a' = 'a ';
+------------+
| 'a' = 'a ' |
+------------+
|          1  |
+------------+
mysql> SET NAMES 'utf8mb4' COLLATE 'utf8mb4_0900_ai_ci';
mysql> SELECT 'a' = 'a ';
+------------+
| 'a' = 'a ' |
+------------+
|          0  |
+------------+
```

The INFORMATION_SCHEMA COLLATIONS table now has a PAD_ATTRIBUTE column that indicates the pad attribute for each collation.

A problem with the latin1_de collation involving early weight string truncation has been corrected. The only likely effect is for WEIGHT_STRING() function results.

- Complementing earlier work in MySQL 8.0.0 to add case-insensitive and accent-insensitive collations for the utf8mb4 Unicode character set, new case-sensitive and accent-sensitive collations have been added. The general collation is named utf8mb4_0900_as_cs, and there are language-specific collations with characteristics similar to utf8mb4_0900_as_cs except that language-specific rules take precedence where applicable. The language-specific collations are indicated by ISO 639-1 language codes in the collation name, as shown in the following table. In two cases the language code has an additional item that denotes a variant (German phone book order, Traditional Spanish).

```
<table>
<thead>
<tr>
<th>Language</th>
<th>Collation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Croatian</td>
<td>utf8mb4_hr_0900_as_cs</td>
</tr>
<tr>
<td>Czech</td>
<td>utf8mb4_cs_0900_as_cs</td>
</tr>
<tr>
<td>Danish</td>
<td>utf8mb4_da_0900_as_cs</td>
</tr>
<tr>
<td>Esperanto</td>
<td>utf8mb4_eo_0900_as_cs</td>
</tr>
<tr>
<td>Estonian</td>
<td>utf8mb4_et_0900_as_cs</td>
</tr>
<tr>
<td>German phone book order</td>
<td>utf8mb4_de_pb_0900_as_cs</td>
</tr>
<tr>
<td>Hungarian</td>
<td>utf8mb4_hu_0900_as_cs</td>
</tr>
</tbody>
</table>
```

Note

utf8mb4_ja_0900_as_cs is the first Japanese language-specific collation available for Unicode in MySQL.
<table>
<thead>
<tr>
<th>Language</th>
<th>Collation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Icelandic</td>
<td>utf8mb4_is_0900_as_cs</td>
</tr>
<tr>
<td>Japanese</td>
<td>utf8mb4_ja_0900_as_cs</td>
</tr>
<tr>
<td>Latvian</td>
<td>utf8mb4_lv_0900_as_cs</td>
</tr>
<tr>
<td>Lithuanian</td>
<td>utf8mb4_lt_0900_as_cs</td>
</tr>
<tr>
<td>Polish</td>
<td>utf8mb4_pl_0900_as_cs</td>
</tr>
<tr>
<td>Classical Latin</td>
<td>utf8mb4_la_0900_as_cs</td>
</tr>
<tr>
<td>Romanian</td>
<td>utf8mb4_ro_0900_as_cs</td>
</tr>
<tr>
<td>Slovak</td>
<td>utf8mb4_sk_0900_as_cs</td>
</tr>
<tr>
<td>Slovenian</td>
<td>utf8mb4_sl_0900_as_cs</td>
</tr>
<tr>
<td>Modern Spanish</td>
<td>utf8mb4_es_0900_as_cs</td>
</tr>
<tr>
<td>Traditional Spanish</td>
<td>utf8mb4_es_trad_0900_as_cs</td>
</tr>
<tr>
<td>Swedish</td>
<td>utf8mb4_sv_0900_as_cs</td>
</tr>
<tr>
<td>Turkish</td>
<td>utf8mb4_tr_0900_as_cs</td>
</tr>
<tr>
<td>Vietnamese</td>
<td>utf8mb4_vi_0900_as_cs</td>
</tr>
</tbody>
</table>

`utf8mb4_0900_as_cs` also works as an accent-sensitive, case-sensitive collation for the languages in the following table.

### Table 3 Languages for Which `utf8mb4_0900_as_cs` is Suitable

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>German (dictionary order)</td>
<td>de</td>
</tr>
<tr>
<td>English</td>
<td>en</td>
</tr>
<tr>
<td>French (locale fr_FR)</td>
<td>fr</td>
</tr>
<tr>
<td>Irish Gaelic</td>
<td>ga</td>
</tr>
<tr>
<td>Indonesian</td>
<td>id</td>
</tr>
<tr>
<td>Italian</td>
<td>it</td>
</tr>
<tr>
<td>Luxembourgian</td>
<td>lb</td>
</tr>
<tr>
<td>Malay</td>
<td>ms</td>
</tr>
<tr>
<td>Dutch</td>
<td>nl</td>
</tr>
<tr>
<td>Portuguese</td>
<td>pt</td>
</tr>
<tr>
<td>Swahili</td>
<td>sw</td>
</tr>
<tr>
<td>Zulu</td>
<td>zu</td>
</tr>
</tbody>
</table>

`utf8mb4_0900_as_cs` is suitable for French French (locale `fr_FR`) but not for Canadian French (locale `fr_CA`). For Canadian French, `utf8mb4_0900_ai_ci` is suitable. The reason for the difference between `ai_ci` and `as_cs` collations is that Canadian French has a different order of accented characters than French French.

`utf8mb4_da_0900_as_cs` also works as an accent-sensitive, case-sensitive collation for the languages in the following table.
Table 4 Languages for Which utf8mb4_da_0900_as_cs is Suitable

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Norwegian</td>
<td>no</td>
</tr>
<tr>
<td>Norwegian Bokmål</td>
<td>nb</td>
</tr>
<tr>
<td>Norwegian Nynorsk</td>
<td>nn</td>
</tr>
</tbody>
</table>

The nonlanguage-specific utf8mb4_0900_as_cs and language-specific utf8mb4_LANG_0900_as_cs Unicode collations each have these characteristics:

- The collation is based on Unicode Collation Algorithm (UCA) 9.0.0 and Common Locale Data Repository (CLDR) v30, is accent sensitive, and case sensitive. These characteristics are indicated by _0900, _as, and _cs in the collation name. Exception: utf8mb4_la_0900_as_cs is not based on CLDR because Classical Latin is not defined in CLDR.

- The collation works for all characters in the range [U+0, U+10FFFF].

- If the collation is not language specific, it sorts all characters, including supplemental characters, in default order (described following). If the collation is language specific, it sorts characters of the language correctly according to language-specific rules, and characters not in the language in default order.

- By default, the collation sorts characters having a code point listed in the DUCET table (Default Unicode Collation Element Table) according to the weight value assigned in the table. The collation sorts characters not having a code point listed in the DUCET table using their implicit weight value, which is constructed according to the UCA.

  The collation sorts on all three DUCET weight levels, including the tertiary level. This compares with accent-insensitive, case-insensitive collations, which sort only on the primary and secondary levels.

- For non-language-specific collations, characters in contraction sequences are treated as separate characters. For language-specific collations, contractions might change character sorting order.

For more information, see Unicode Character Sets.

Compilation Notes

- **Incompatible Change:** The my_bool type is no longer used in MySQL source code. Any third-party code that used this type to represent C boolean variables should use the bool or int C type instead.

  ![Note]

  The change from my_bool to bool means that the mysql.h header file now requires a C++ or C99 compiler to compile.

  (Bug #25597667)

- **InnoDB:** A LOB page deletion function failed to delete LOB pages. (Bug #24480254)

- Windows builds now use the default runtime libraries (builds use the /MD flag). (Bug #25611609)

- **CMake** support was added for compiling with Developer Studio 12.6. (Bug #25384295)

- The rapidjson library included in MySQL distributions was upgraded to version 1.1.0. (Bug #24947436, Bug #83515)
• The minimum version of the Boost library for server builds is now 1.63.0. (Bug #24579061, Bug #82834, Bug #25126144, Bug #83905)

• For GCC versions higher than 4.4, -fno-expensive-optimizations was replaced with -ffp-contract=off, which has the effect of enabling more optimizations. Thanks to Alexey Kopytov for the patch. (Bug #24571672, Bug #82760)

• For building MySQL 8.0, the minimum required version of CMake is now 3.2.3 on Windows, 3.4.0 on Solaris, and 2.8.12 otherwise. (Bug #24481181, Bug #82628)

References: See also: Bug #24687701.

• Work was done to clean up the source code base, including: Removing unneeded CMake checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

**Configuration Notes**

• In source distributions, several utilities previously in the extra directory have been moved to the new utilities directory. (Bug #25416084)

• Two new CMake options are available for debugging. LINK_RANDOMIZE indicates whether to randomize the order of symbols in the mysqld binary (default OFF), and LINK_RANDOMIZE_SEED specifies a seed value for LINK_RANDOMIZE. (Bug #25336715)

• MySQL failed to compile if -DENABLE_DEBUG_SYNC=OFF AND -DWITH_DEBUG=ON were both given. The ENABLE_DEBUG_SYNC option has been removed and enabling WITH_DEBUG enables Debug Sync. (Bug #18374703)

• These changes were made with respect to persisted system variables:

  • A new statement, RESET PERSIST, enables removal of variable settings that were persisted with SET PERSIST to the mysqld-auto.cnf file that stores persisted global system variable settings.

  • Using SET PERSIST (or @@PERSIST.) to set a global variable to DEFAULT or to the variable literal default value previously also added a setting for the variable to the mysqld-auto.cnf file if was is not present, or removed it from mysqld-auto.cnf if it was present. Now the assignment always adds a setting for the variable to the mysqld-auto.cnf file; to remove the setting from the file, use RESET PERSIST.

  • A new Performance Schema table, persisted_variables, provides an SQL interface to the mysqld-auto.cnf file, enabling the file contents to be inspected at runtime using SELECT statements.

  • The Performance Schema variables_info table has new columns showing when and by which user each system variable was most recently set.

For more information, see Persisted System Variables, and RESET PERSIST Syntax.

If you upgrade to this MySQL release from an earlier version, you must run mysql_upgrade (and restart the server) to incorporate these changes into the Performance Schema.

References: See also: Bug #24522064.

• The default value of the query_cache_size system variable has been reduced from 1M to 0. Consequently no query cache buffer is allocated by default.
Data Dictionary Notes

- SQL-layer code and the storage engine API were extended to provide data dictionary support for:
  - Storing information about auxiliary columns and keys that InnoDB adds to tables implicitly, storage engine-private data and object identifiers for tables and tablespaces, and InnoDB full-text search auxiliary tables and other similar, implicitly created objects.
  - Combining data dictionary updates, storage engine changes, and binary log writes for DDL operations into atomic transactions.
  
A behavior change for DROP TABLE occurs if any named tables do not exist. Previously, the statement returned an error indicating which tables did not exist and it was unable to drop, but also dropped the tables that did exist. Now the statement still indicates which tables did not exist, but fails with an error and no changes are made.

For more information, see Atomic Data Definition Statement Support, and WL#7743.

- These INFORMATION_SCHEMA tables have been reimplemented as views on data dictionary tables:

  EVENTS
  PARAMETERS
  ROUTINES
  TRIGGERS

Queries on those tables are now more efficient because they obtain information from data dictionary tables rather than by other, slower means. For example, the server no longer must create a temporary table for each query of the INFORMATION_SCHEMA table.

Improvements for those tables also apply to SHOW statements that display information corresponding to the INFORMATION_SCHEMA tables. For example, SHOW TRIGGERS displays the same information as the TRIGGERS table.

If you upgrade to this MySQL release from an earlier version, you must run mysql_upgrade (and restart the server) to incorporate these changes.

Deprecation and Removal Notes

- Incompatible Change; InnoDB; Partitioning: The ALTER TABLE ... UPGRADE PARTITIONING statement is no longer supported.
- The deprecated PROCEDURE ANALYSE() syntax has been removed.
- The deprecated libmysqld embedded server library has been removed, along with:
  - The mysql_config --libmysqld-libs, --embedded-libs, and --embedded options
  - The CMake WITH_EMBEDDED_SERVER, WITH_EMBEDDED_SHARED_LIBRARY, and INSTALL_SECURE_FILE_PRIV_EMBEDDEDDIR options
  - The (undocumented) mysql --server-arg option
  - The mysqltest --embedded-server, --server-arg, and --server-file options
  - The mysqltest_embedded and mysql_client_test_embedded test programs
- The deprecated --temp-pool server option has been removed.
• Support for DTrace has been removed.

**Optimizer Notes**

• **InnoDB**: MySQL now supports descending indexes: *DESC* in an index definition is no longer ignored but causes storage of key values in descending order. Previously, indexes could be scanned in reverse order but at a performance penalty. A descending index can be scanned in forward order, which is more efficient. Descending indexes also make it possible for the optimizer to use multiple-column indexes when the most efficient scan order mixes ascending order for some columns and descending order for others. For more information, see [Descending Indexes](#).

**Note**

Previously, `GROUP BY` implicitly sorted by default (that is, in the absence of `ASC` or `DESC` designators for `GROUP BY` columns), but relying on implicit `GROUP BY` sorting was deprecated. `GROUP BY` no longer sorts by default, so query results may differ from previous MySQL versions. To produce a given sort order, use explicit `ASC` or `DESC` designators for `GROUP BY` columns or provide an `ORDER BY` clause.

• Optimizer trace output now includes more information about `filesort` operations, such as key and payload size and why addon fields are not packed. (Bug #25246184, Bug #84180)

• Previously, invisible indexes were supported only for the `InnoDB` storage engine. Invisible indexes are now storage engine neutral (supported for any engine). (Bug #23541244)

• A `GROUPING()` function has been introduced for use in `GROUP BY` queries that include a `WITH ROLLUP` modifier. This function enables you to distinguish `NULL` values for super-aggregate rows in the query result from `NULL` values in regular grouped rows. For more information, see [Miscellaneous Functions](#) and `GROUP BY Modifiers`. Thanks to Zhe Dong for a patch that was partially used to implement this feature.

**Note**

`GROUPING` now is a reserved word and cannot be used as an identifier without identifier quoting.

(Bug #11754449, Bug #46053, Bug #11745963, Bug #21728)

• The optimizer now supports `INDEX_MERGE` and `NO_INDEX_MERGE` hints to affect use of the Index Merge access method. Examples:

```sql
SELECT /*+ INDEX_MERGE(t1 f1, f2, f3) */ f2 FROM t1
WHERE f1 = 'o' AND f2 = f3 AND f3 <= 4;
SELECT /*+ NO_INDEX_MERGE(t1 f2, f3) */ f1 FROM t1
WHERE (f2 = 5 OR f3 = 'c') AND (f1 = 4 OR f4 = 'f');
```

For more information, see [Index Merge Optimization](#) and [Optimizer Hints](#).

• The `server_cost` and `engine_cost` optimizer cost model tables in the `mysql` system database now include a `default_value` column that contains the default value for each cost table estimate. This column is a read-only generated column that retains its value even if the associated cost estimate is changed. For more information, see [The Optimizer Cost Model](#).

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `mysql` system database.
• In the optimizer cost model `engine_cost` table, the cost for memory access was decreased relative to the cost for disk access. An implication of this change is that the execution plan may change between two runs of the same query. For example, at server startup before data has been read into the buffer pool, you may get a different plan than after the query has been run because then the data will be in memory.

• MySQL now supports common table expressions, both nonrecursive and recursive. Common table expressions enable use of named temporary result sets, implemented by permitting a `WITH` clause preceding `SELECT` statements and certain other statements. For more information, see WITH Syntax (Common Table Expressions).

  Note

  `RECURSIVE` now is a reserved word and cannot be used as an identifier without identifier quoting.

• The optimizer now supports hints that enable specifying the order in which to join tables. For more information, see Optimizer Hints.

Packaging Notes

• Microsoft Windows: Reminder: MySQL 8.0 requires the Microsoft Visual C++ 2015 Redistributable Package to run on Windows platforms. Users should make sure the package has been installed on the system before starting the server. The package is available at the Microsoft Download Center.

• Changes in RPM package structure require a larger set of packages to be removed to install MySQL Server cleanly. (Bug #25603087)

• To avoid potential race conditions, Debian packages now use the GNU `install` utility rather than a combination of `mkdir`, `touch`, and `chown`. (Bug #25258829)

• New Debian/Ubuntu packages named `server-core` and `client-core` are now available. These packages contain binaries only, without configuration or service scripts, directory setup, man pages, and so forth. They are installed automatically with the standard packages, but when installed by themselves do not result in a functioning MySQL setup. They may be of interest for user which wish to customize their own installation setup. (Bug #25146364)

• `scripts/mysql_security_commands.sql` and `support-files/mysql.server-sys5.sh` are no longer used and have been removed from MySQL distributions. (Bug #24756400, Bug #24756442)

• RPM packages now are built with `-DWITH_NUMA=ON` for platforms with NUMA support: OEL higher than EL5, Fedora, SLES, Docker. (Bug #24689078)

• The `my-default.cnf.sh` file (used to produce a default `my-default.cnf` or `my-default.ini` file) is no longer included in source distributions and `my-default.cnf` and `my-default.ini` are no longer included in or installed by distribution packages. (Bug #22525354)

• The unused and undocumented `archive_reader` and `archive_test` utilities have been removed from MySQL source distributions. (Bug #12818207, Bug #62014)

Parser Notes

• The parser no longer considers `\N` as a synonym for `NULL` in SQL statements. Use `NULL` instead.

  This change does not affect text file import or export operations performed with `LOAD DATA` or `SELECT ... INTO OUTFILE`, for which `NULL` continues to be represented by `\N`. See LOAD DATA Syntax.
Performance Schema Notes

- The Performance Schema maintains statement event summary tables that contain information about minimum, maximum, and average statement latency. Those tables permit high-level assessment of system performance. To permit assessment at a more fine-grained level, the Performance Schema now also collects histogram data for statement latencies. These histograms provide additional insight into latency distributions and are available in these tables:

  - `events_statements_histogram_by_digest`: Statement histograms per schema and digest value
  - `events_statements_histogram_global`: Statement histogram summarized globally

  For more information, see Statement Histogram Summary Tables.

  In addition, the `events_statements_summary_by_digest` table now has columns that provide percentile information about statement latencies. For more information, see Statement Summary Tables.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the Performance Schema.

- System and status variable information is no longer maintained in the INFORMATION_SCHEMA. These tables have been removed: `GLOBAL_VARIABLES`, `SESSION_VARIABLES`, `GLOBAL_STATUS`, `SESSION_STATUS`. Use the corresponding Performance Schema tables instead. See Performance Schema System Variable Tables, and Performance Schema Status Variable Tables. In addition, the `show_compatibility_56` system variable has been removed. It was used in the transition period during which system and status variable information in INFORMATION_SCHEMA tables was moved to Performance Schema tables, and is no longer needed. These status variables have been removed: `Slave_heartbeat_period`, `Slave_last_heartbeat`, `Slave_received_heartbeats`, `Slave_retried_transactions`, `Slave_running`. The information they provided is available in Performance Schema tables; see Migrating to Performance Schema System and Status Variable Tables.

- The deprecated INFORMATION_SCHEMA `INNODB_LOCKS` and `INNODB_LOCK_WAITS` tables have been removed. To obtain InnoDB data lock information, use the Performance Schema `data_locks` and `data_lock_waits` tables instead. Alternatively, use the `sys` schema `innodb_lock_waits` and `x$innodb_lock_waits` views, which have been reimplemented to use the new Performance Schema tables rather than the removed INFORMATION_SCHEMA tables.

  The Performance Schema tables expose what data locks exist, which transaction holds each lock, and which transaction lock requests are blocked by other locks. For InnoDB, the `data_locks` table reports all data locks, in contrast to the `INNODB_LOCKS` table, which reported only waited-for data locks. For more information, see The data_locks Table, and The data_lock_waits Table. For information about mapping old table columns to new table columns, see The INFORMATION_SCHEMA INNODB_LOCKS Table, and The INFORMATION_SCHEMA INNODB_LOCK_WAITS Table.

  The tables differ in the privileges required: The INFORMATION_SCHEMA tables require the global PROCESS privilege. The Performance Schema tables require the usual Performance Schema privilege of SELECT on the table to be selected from.

  The `sys` schema `innodb_lock_waits` and `x$innodb_lock_waits` views can be used without change, with the exception that the `locked_table` column containing combined schema/table name values has been removed and replaced with `locked_table_schema` and `locked_table_name` columns that contain separate schema and table name values. In addition, these views now contain
locked_table_partition and locked_table_subpartition columns. For more information, see The innodb_lock_waits and x$innodb_lock_waits Views.

If you upgrade to this MySQL release from an earlier version, you must run mysql_upgrade (and restart the server) to incorporate these changes into the Performance Schema and the sys schema.

Plugin Notes

- MySQL Server 8.0 includes a component-based infrastructure for improving server extensibility; see MySQL Server Components. However, MySQL plugins use an interface that predates the component interface. A new plugin service, plugin_registry_service, enables plugins to access the component registry and its services.

Security Notes

- **Incompatible Change:** For STANDALONE and WIN builds, the default secure_file_priv value has changed from the empty string to NULL. This is a secure-by-default setting because it disables import and export operations. To permit those operations, set secure_file_priv to the path name of the directory to use for those operations. (Bug #24679907, Bug #24695274, Bug #24707666)

- **Incompatible Change:** These changes were made to mysqld_safe:
  - Unsafe use of rm and chown in mysqld_safe could result in privilege escalation. chown now can be used only when the target directory is /var/log. An incompatible change is that if the directory for the Unix socket file is missing, it is no longer created; instead, an error occurs. Due to these changes, /bin/bash is required to run mysqld_safe on Solaris. /bin/sh is still used on other Unix/Linux platforms.
  - The --ledir option now is accepted only on the command line, not in option files.
  - mysqld_safe ignores the current working directory.

Other related changes:

- Initialization scripts that invoke mysqld_safe pass --basedir explicitly.
- Initialization scripts create the error log file only if the base directory is /var/log or /var/lib.
- Unused systemd files for SLES were removed.

(Bug #24483092, Bug #25088048, Bug #25378439, Bug #25378565)

References: See also: Bug #24464380, Bug #24388753, Bug #24619033, Bug #82920.

- **Incompatible Change:** Server components and plugins now can define privileges dynamically (at runtime), which enables them to create privileges specifically associated with the capabilities they implement.

Implementation of dynamic privileges enables DBAs to begin migrating away from the SUPER privilege. In the past, SUPER has been associated with a wide and growing variety of operations, not all related to each other. Many operations covered by SUPER now are also associated with a dynamic privilege of more limited scope, and that privilege is registered by the server component or plugin that implements the operation. Each such operation that previously required the SUPER privilege can be permitted to an account by granting the associated dynamic privilege rather than SUPER. For example, a user who must be able to modify global system variables can be granted SYSTEM_VARIABLES_ADMIN rather than SUPER.
This change improves security by enabling DBAs to avoid granting `SUPER` and tailor user privileges more closely to the operations permitted. `SUPER` is now deprecated and will be removed in a future MySQL version.

Dynamic privileges apply only at the global level. The server stores information about current assignments of dynamic privileges to user accounts in `mysql.global_grants`, a new table in the `mysql` system database.

For more information, see Static Versus Dynamic Privileges. That discussion includes instructions for migrating accounts away from `SUPER` to dynamic privileges.

Incompatibility: `SHOW GRANTS` no longer displays `ALL PRIVILEGES` in its global-privileges output because the meaning of `ALL PRIVILEGES` at the global level varies depending on which dynamic privileges are defined. Instead, `SHOW GRANTS` explicitly lists each granted global privilege. For details, see SHOW GRANTS Syntax. Applications that process `SHOW GRANTS` output should be adjusted accordingly.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `mysql` system database.

- yaSSL was upgraded to version 2.4.2. This upgrade corrects issues with: Potential AES side channel leaks; DSA padding for unusual sizes; the `SSL_CTX_load_verify_locations()` OpenSSL compatibility function failing to handle long path directory names. (Bug #24512715, Bug #24740291)

- MySQL Server now includes a plugin library that enables administrators to introduce an increasing delay in server response to clients after a certain number of consecutive failed connection attempts. This capability provides a deterrent that slows down brute force attacks that attempt to access MySQL user accounts. For more information, see The Connection-Control Plugins.

- OpenSSL is ending support for version 1.0.1 in December 2016; see https://www.openssl.org/policies/releasestrat.html. Consequently, MySQL Commercial Server builds now use version 1.0.2 rather than version 1.0.1, and the linked OpenSSL library for the MySQL Commercial Server has been updated from version 1.0.1 to version 1.0.2j. For a description of issues fixed in this version, see https://www.openssl.org/news/vulnerabilities.html.

This change does not affect the Oracle-produced MySQL Community build of MySQL Server, which uses the yaSSL library instead.

### Spatial Data Support

- **Incompatible Change:** Previously, `ST_Distance()` supported only Cartesian spatial reference systems (SRSs). `ST_Distance()` now detects geometry arguments in a geographic (ellipsoidal) SRS and returns the geodetic distance on the ellipsoid in meters. The only permitted geographic argument types are `Point` and `Point`, or `Point` and `MultiPoint` (in any argument order). Calculations for projected SRSs and SRID 0 remain the same. For more information, see Spatial Relation Functions That Use Object Shapes.

  **Note**

  If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

- **Incompatible Change:** These functions previously accepted either WKB strings or geometry arguments. Geometry arguments are no longer permitted and produce an error.
MySQL 8.0 Release Notes

- `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromWKB()`
- `ST_GeomFromWKB()`, `ST_GeometryFromWKB()`
- `ST_LineFromWKB()`, `ST_LinestringFromWKB()`
- `ST_MLineFromWKB()`, `ST_MultiLinestringFromWKB()`
- `ST_MPointFromWKB()`, `ST_MultiPointFromWKB()`
- `ST_MPolyFromWKB()`, `ST_MultiPolygonFromWKB()`
- `ST_PointFromWKB()`
- `ST_PolyFromWKB()`, `ST_PolygonFromWKB()`

For information about migrating queries that refer to those functions away from using geometry arguments to using WKB arguments, see Functions That Create Geometry Values from WKB Values.

- **Incompatible Change:** These functions now interpret latitude and longitude coordinates as in the order specified by the spatial reference system. The functions also accept an optional argument to override the default axis order.
  - `ST_GeomCollFromText()`, `ST_GeometryCollectionFromText()`, `ST_GeomCollFromTxt()`
  - `ST_GeomFromText()`, `ST_GeometryFromText()`
  - `ST_LineFromText()`, `ST_LinestringFromText()`
  - `ST_MLineFromText()`, `ST_MultiLinestringFromText()`
  - `ST_MPointFromText()`, `ST_MultiPointFromText()`
  - `ST_MPolyFromText()`, `ST_MultiPolygonFromText()`
  - `ST_PointFromText()`
  - `ST_PolyFromText()`, `ST_PolygonFromText()`
  - `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromWKB()`
  - `ST_GeomFromWKB()`, `ST_GeometryFromWKB()`
  - `ST_LineFromWKB()`, `ST_LinestringFromWKB()`
  - `ST_MLineFromWKB()`, `ST_MultiLinestringFromWKB()`
  - `ST_MPointFromWKB()`, `ST_MultiPointFromWKB()`
  - `ST_MPolyFromWKB()`, `ST_MultiPolygonFromWKB()`
  - `ST_PointFromWKB()`
  - `ST_PolyFromWKB()`, `ST_PolygonFromWKB()`

For more information, see Functions That Create Geometry Values from WKT Values, and Functions That Create Geometry Values from WKB Values.
Note
If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

• **Incompatible Change:** These functions now return latitude and longitude coordinates in the order specified by the spatial reference system that applies to the geometry value argument. The functions also accept an optional argument to override the default axis order.
  
  • `ST_AsBinary()`, `ST_AsWKB()`
  • `ST_AsText()`, `ST_AsWKT()`

For more information, see [Geometry Format Conversion Functions](#).

Note
If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

• The `st_spatial_reference_systems` data dictionary table that stores information about spatial reference systems other than SRID 0 has been upgraded from EPSG Dataset 8.7 to 9.0. Also, the table contents are now exposed through the `INFORMATION_SCHEMA ST_SPATIAL_REFERENCE_SYSTEMS` table, which is implemented as a view on the data dictionary. For more information, see [The INFORMATION_SCHEMA ST_SPATIAL_REFERENCE_SYSTEMS Table](#).

MySQL now has an `INFORMATION_SCHEMA ST_GEOMETRY_COLUMNS` table that provides information about table columns that store spatial data. The new table is implemented as a view on the existing `INFORMATION_SCHEMA COLUMNS` table. For more information, see [The INFORMATION_SCHEMA ST_GEOMETRY_COLUMNS Table](#).

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these `INFORMATION_SCHEMA` changes. (Bug #25337054, Bug #84384)

• A new spatial function, `ST_SwapXY()`, accepts a geometry argument and swaps the X and Y values of each coordinate pair within the geometry. For more information, see [Geometry Format Conversion Functions](#).

### SQL Syntax Notes

• Derived table syntax now permits a list of explicit column names following the table name, to override the default column names taken from the select list. For example, the column names for this derived table come from its select list:

```
mysql> SELECT * FROM (SELECT 1, 2, 3, 4) AS dt;
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
```

To provide column names, follow the derived table name with a parenthesized list of column names:

```
mysql> SELECT * FROM (SELECT 1, 2, 3, 4) AS dt (a, b, c, d);
```
Test Suite Notes

- `mysql-test-run.pl` now has a `--charset-for-testdb=charset_name` option for specifying the default character set for the test database. The default value is `latin1`. (Bug #25494068, Bug #84806)

- `mysql-test-run.pl` now has an `--only-big-tests` option, which causes only big tests to run. Normal (non-big) tests are skipped. If both `--big-test` and `--only-big-tests` are given, `--only-big-tests` is ignored. (Bug #25182306)

- `mysql-test-run.pl` has a new `--discover` option. When given, `mysql-test-run.pl` attempts to preload `discover`, the Developer Studio Memory Error Discovery Tool when starting `mysqld`. Reports from `discover` may be found in `log/mysqld.%p.txt` under the directory given by `--vardir`. This option is supported only on SPARC-M7 systems. (Bug #25048971)

- `mysqltest` now supports an `expr` command that enables an expression to be evaluated and assigned to a variable. For details, see the command description in the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html. (Bug #24806741)

- `mysqltest` now supports a `force-cpdir` command that copies a source directory to a destination directory recursively (that is, it copies subdirectories). (Bug #24806681)

- `mysql-test-run.pl` has a new `--summary-report` option. When given, `mysql-test-run.pl` generates a plain text version of the test summary only and writes it to the file named as the option argument. The file is suitable for sending by email. (Bug #24512357, Bug #82708)

- `mysql-test-run.pl` could not be run with `--valgrind-option=--tool=custom_tool`, for values of `custom_tool` such as `massif` or `helgrind`, because it added the options for `memcheck` that might not be understood by other tools. Also, the `mysql-test-run.pl --callgrind` option did not work because it supplied an invalid `--base` option to `callgrind`. Thanks to Daniel Black for the patch on which the fixes were based. (Bug #23713613, Bug #82039)

- The `mysqltest connect()` function now permits `SOCKET` and `TCP` values for the `options` argument, to specify using the socket-file and TCP/IP connection protocols.

In addition, connection protocols now must match the current system. Previously, passing `PIPE` or `SHM` on non-Windows systems caused the default (socket-file) connection protocol to be used. Now this causes an error, and, similarly, passing `SOCKET` on Windows systems causes an error. (Bug #21046241)

- `mysql-test-run.pl` now supports a `--bootstrap` option that enables `mysqld` options to be designated as bootstrap options. When `mysql-test-run.pl` finds `--bootstrap` options in a `master.opt` file, it removes and reinitializes the data directory and restarts the server with the new option settings. (Bug #18184868)

X Plugin Notes

- The `MYSQL_TCP_PORT` CMake option was ignored. (Bug #25493867, Bug #84804)

- The protobuf compiler emitted "unused import" warnings for `mysqlx.proto`. (Bug #25444009, Bug #84641)

- Connections were released which were not initialized when validation occurred. (Bug #25392280)
• The `Mysqlx_notice_warning_sent` and `Mysqlx_notice_other_sent` status variable values did not increment as expected. (Bug #25289949, Bug #84258, Bug #25290001, Bug #84260)

• The manual definition of the `MYSQL_DYNAMIC_PLUGIN` macro was redundant. (Bug #25162590, Bug #83988)

• IPv6 connectivity support defined a new system variable `mysqlx_bind_address` with the default value of '0.0.0.0', but the default value should have been '*' so that connections to X Plugin may be made using an IPv6 address, an IPv4 address, or a hostname. (Bug #25047909, Bug #83688)

• The `stmt` field was marked as required in the message:

```plaintext
message Mysqlx.Crud.ModifyView {
  required Collection collection = 1;
  optional string definer = 2;
  optional ViewAlgorithm algorithm = 3;
  optional ViewSqlSecurity security = 4;
  optional ViewCheckOption check = 5;
  repeated string column = 6;
  required Mysqlx.Crud.Find stmt = 7;
}
```

(Bug #24968735, Bug #83595)

• For a table created with `CREATE TABLE foo (doc JSON)`, the statement `list_objects` incorrectly reported the table as a collection. (Bug #24963952)

• Sending `Expect.Open` to a connected socket before authentication led to the following error messages:

```plaintext
... Plugin mysqlx reported: '10: Invalid message 24 received during client initialization'
... Plugin mysqlx reported: '10: ERROR reading from socket Bad file descriptor (9) 2'
```

(Bug #24940144, Bug #83494)

• On Linux, MySQL Shell in JavaScript and Python interactive mode hid collections if the letter case in the collection name was not uniform. (Bug #24848125)

• If multiple user accounts exist that match the authenticating user, the X Plugin would attempt to authenticate all matching accounts instead of the best matching account. (Bug #24847537)

• Boost classes with corresponding functionality in C++ 11 were moved or wrapped. (Bug #24680856)

• When two X Plugin instances were started on the same port, the resulting error message was split over two lines and contained duplicate information. (Bug #24679018, Bug #83008)

• The `performance_schema.threads` table did not show `processlist` statistics for the X Plugin. (Bug #24638038)

• If the client sent an unknown message type to the server, the server would send back an error message and then disconnect the client. (Bug #24611754, Bug #82868)

• When the client sent a zero length message, the X Plugin did not send a response. For a subsequent message the client received an error `E_X_BAD_MESSAGE` and was disconnected, depending on the header content. (Bug #24595459, Bug #82862)

• X Plugin displayed its version to any connected user, including those not authenticated. (Bug #24562707, Bug #82784)
• X Plugin generated an incorrect query when a duplicate field name existed in an `update` statement. (Bug #24510083)

• The statement `list_objects` returned incorrect information for some views in `sys` database. (Bug #24472325)

• X Plugin was trying to send result sets to a client, even where the connection had been closed. The plugin now stops execution of a stored procedure if the connection is reset. (Bug #24440344)

• Attempting to list objects using the `list_objects` statement without explicitly setting the database name resulted in the collection being incorrectly classified as a table. (Bug #23205895, Bug #81248)

• X Plugin now supports the `sha256_password` plugin.

**Platform-Specific Notes**

• **Solaris**: The minimum required version of Solaris is now Solaris 11 update 3, due to a dependency on system runtime libraries.

• **Solaris**: On Solaris, MySQL is now built with Developer Studio 12.5 instead of `gcc`. The binaries require the Developer Studio C/C++ runtime libraries to be installed. See here for how to install only the libraries:

  https://docs.oracle.com/cd/E60778_01/html/E60743/gozsu.html

**Functionality Added or Changed**

• **Incompatible Change**: Previously, for `insert ... select ... on duplicate key update` statements for which the `update` referred to `select` columns, the results could be incorrect if the `select` was a `union`. Now such statements are prohibited and produce an error. To work around this restriction, write the `union` as a derived table so that its rows can be treated as a single-table result set, which is permitted. See `insert ... on duplicate key update` syntax.

• **InnoDB**: By default, InnoDB reads uncommitted data when calculating statistics. In the case of an uncommitted transaction that deletes rows from a table, InnoDB excludes records that are delete-marked when calculating row estimates and index statistics, which can lead to non-optimal execution plans for other transactions that are operating on the table concurrently using a transaction isolation level other than `read uncommitted`. To avoid this scenario, a new configuration option, `innodb_stats_include_delete_marked`, can be enabled to ensure that InnoDB includes delete-marked records when calculating persistent optimizer statistics. (Bug #23333990)

• **InnoDB**: Geometry parsing and bounding box computational code for R-trees was moved from InnoDB to the server.

• **InnoDB**: InnoDB now supports `nowait` and `skip locked` options with `select ... for share` and `select ... for update` locking read statements. `nowait` causes the statement to return immediately if a requested row is locked by another transaction. `skip locked` removes locked rows from the result set. See Locking Read Concurrency with NOWAIT and SKIP LOCKED.

  `select ... for share` replaces `select ... lock in share mode`, but `lock in share mode` remains available for backward compatibility. The statements are equivalent. However, `for share` supports `nowait`, `skip locked`, and of `tbl_name` options. See `select syntax`.

  OF `tbl_name` applies locking queries to named tables.
Of now is a reserved word and cannot be used as an identifier without identifier quoting.

- **InnoDB:** The InnoDB data-at-rest encryption feature now supports encryption of redo log and undo log data, controlled by the `innodb_redo_log_encrypt` and `innodb_undo_log_encrypt` configuration options. See Redo Log Encryption, and Undo Log Encryption.

- **InnoDB:** InnoDB internal temporary tables that are stored on disk now support multiple cursor positions, permitting single writer and multiple reader access within the same thread. The purpose of this enhancement is to provide support for recursive and non-recursive common table expressions (CTEs).

  Additionally, consistent-read access to InnoDB on-disk internal temporary tables is replaced by a dirty-read scheme, and row counts for InnoDB on-disk internal temporary tables now use row count statistics instead of slower table scans.

- **InnoDB:** InnoDB now compresses large objects into a sequence of smaller zlib streams for tables that use `ROW_FORMAT=COMPRESSED`. Previously, large object data was compressed into a single zlib stream.

- **Replication:** View change events from a Group Replication group can now be replicated to an external multithreaded slave (MTS) of type DATABASE. (Bug #25170698, Bug #84008)

- **Replication:** When a negative or fractional timeout parameter was supplied to `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()`, the server behaved in unexpected ways. With this fix:

  - A fractional timeout value is read as-is, with no round-off.
  - A negative timeout value is rejected with an error if the server is on a strict SQL mode; if the server is not on a strict SQL mode, the value makes the function return NULL immediately without any waiting and then issue a warning.

  (Bug #24976304, Bug #83537)

- **Replication:** Work has been done to reduce contention between receiver (I/O) and applier (SQL) replication threads. This ensures slaves keep up with a higher insert load on their master. (Bug #78246, Bug #21753696)

- **Replication:** Added the `binlog_expire_logs_seconds` system variable, which sets an interval in seconds for purging of the binary log. The effects of this variable and `expire_logs_days` are cumulative, making it possible to set a period such as 1.5 days. To completely disable automatic binary log purging, set both variables equal to 0, which is the default value for both of them. (Bug #71697, Bug #18260088)

- **Replication:** Performance schema tables have been added to monitor replication lags and queues. The `replication_connection_status` table has updated information on the last transaction queued in the relay log, as well as the transaction currently being queued in the relay log. The `replication_applier_status_by_coordinator` table has updated information on the last transaction written to the buffer of a worker, as well as the transaction currently being processed by the coordinator. The `replication_applier_status_by_worker` table has updated information on the last transaction applied by the worker, as well as on the transaction currently being applied. The information presented in each of these tables contains the transaction’s GTID, commit timestamps, processing stage timestamp, and the timestamp of the completion.
In the `replication_applier_status_by_worker` table, the `LAST_SEEN_TRANSACTION` column is replaced by the `APPLYING_TRANSACTION` column.

- **Replication:** The new `binlog_expire_logs_seconds` variable is a global server variable in addition to the existing `expire_logs_days` variable. The purpose is to facilitate finer grained retention policies of binary logs than the current day granularity.

- **Replication:** Correct recovery of DDL statements or transactions by binary log implements support for correct recovery of DDL transactions based on the binary log data. In the context of the new data-dictionary it becomes possible to make metadata changes and write them to the binary log as a single transaction by using the 2-phase commit protocol.

- **Replication:** Multi-source replication now supports per-channel replication filters. This makes it possible to filter out the execution of selected replicated data from a specific channel. Until now replication filters have been global, and only applicable to all channels.

- **Replication:** Row-based replication now uses extended table metadata in the binary log. The extended metadata serves two major purposes: slaves use the metadata to transfer data smoothly when its table structure is different from master's, and external software can use the metadata to decode row events and store the data into external databases.

- **Replication:** Group Replication now supports `SAVEPOINT` SQL transactions.

- **Replication:** An infrastructure for GTID based delayed replication and replication lag monitoring has been added to enable you to properly monitor replication lag. Two new timestamps that are associated with each transaction (not each event or statement) in the binary log have been introduced. The `original_commit_timestamp` is in microseconds since the epoch when the transaction was committed on the original master, and the `immediate_commit_timestamp` is in microseconds since the epoch when the transaction was committed on the immediate master.

- **Replication:** The `RESET MASTER` statement has been extended to allow specification of a binary log file number. The `RESET MASTER TO` statement accepts an argument which specifies the index of the new binary log file to use. The purpose is to simplify failover procedures by replacing the `FLUSH BINARY LOGS` and `PURGE BINARY LOGS TO` statements with a single `RESET MASTER` statement.

- **Replication:** The process used for how delay is defined and calculated for delayed replication have been improved. This relies on a new timestamp in the binary log, the `immediate_commit_timestamp`, which is the number of microseconds since epoch when the transaction was written to the binary log of the immediate master. This means that the delay is no longer computed per-event, it is now applied per-transaction.

- **Replication:** It is now possible to specify whether information written into the binary log enables replication slaves to parallelize based on commit timestamps, or on transaction write sets. Using write sets has a the potential for greater parallelism than using commit timestamps since it does not depend on the commit history. When applying binary logs in this fashion on a replication slave, it may be able to leverage capabilities of the underlying computing hardware (such as CPU cores) and thus speed up this process.

The interface for choosing the source of parallelization is implemented as a new server system variable `binlog_transaction_dependency_tracking` which can take any one of the values `COMMIT_ORDER`, `WRITESET`, or `WRITESET_SESSION.COMMIT_ORDER` (the default) causes parallelization information to be logged using commit timestamps; `WRITESET` causes this information to be logged using write sets in such a way that any transactions not updating the same row can be parallelized; and `WRITESET_SESSION` acts in the same fashion as `WRITESET`, except that updates originating with the same session cannot be reordered. The size of the
row hash history that is kept in memory for tracking transaction dependencies can be set using
`binlog_transaction_dependency_history_size`, also introduced in this release.

- **JSON**: Added the JSON utility function `JSON_PRETTY()`, which prints an existing JSON value, or any
  string that can successfully be parsed as a JSON document, in a format that can be easily read by
  humans. Each JSON object member or array value is displayed on a separate line of the output; each
  child object or array is intended 2 spaces with respect to its parent.

Examples:

```
mysql> SELECT JSON_PRETTY('123');
+--------------------+
| JSON_PRETTY('123') |
| 123                |
+--------------------+

mysql> SELECT JSON_PRETTY('[1,3,5]');
+------------------------+
| JSON_PRETTY('[1,3,5]') |
| [1, 3, 5]              |
+------------------------+

mysql> SELECT JSON_PRETTY('{"a":"10","b":"15","x":"25"}');
+---------------------------------------------+
| JSON_PRETTY('{"a":"10","b":"15","x":"25"}') |
| { "a": "10", "b": "15", "x": "25" }         |
+---------------------------------------------+
```

- **JSON**: Previously, when sorting JSON values in a query using `ORDER BY`, each value was represented
  by a part having a fixed length in the sort key, each such part requiring 1K of memory. In many cases
  this usage was excessive—for example, an integer scalar value requires only a relatively very few bytes,
  so that the remainder of the 1K (90% or more) was taken up by padding.

Variable length parts for JSON sort keys have been introduced to address this issue, with each key
part now taking up only the space that is needed to store the value. This has the following benefits for
performance:

- Because sort buffer space is used more effectively, filesorts need not flush to disk as early or often,
  which means that more data can be sorted in memory.

- Shorter keys can be compared more quickly than longer ones, providing a noticeable improvement in
  performance for in-memory sorts as well as sorts requiring disk usage.

- The `WEIGHT_STRING()` debugging function no longer supports the `LEVEL` clause. (Bug #25469683,
  Bug #84723)

- For Developer Studio 12.5, 32-bit builds are now disabled unless the `FORCE_UNSUPPORTED_COMPILER`
  option is given. (Bug #25267157, Bug #84230)
• Several foreign key constraint checks that were storage engine agnostic were being done inside InnoDB. These are now done at the SQL layer. (Bug #25252847)

• Some GIS out-of-bounds checking was simplified. Thanks to Daniel Black for the contribution. (Bug #25202470, Bug #84062)

• The set of compiler flags used for Developer Studio 12.5 was improved. (Bug #25148549, Bug #83955)

• CMake now uses CMAKE_POLICY of CMP0022 NEW rather than CMP0022 OLD. (Bug #25090147, Bug #83804)

• If MySQL was configured to build with the -Wno-error option, mysql_config produced incorrect output for its --cflags option. The set of compiler options that mysql_config and pkg-config produce now is determined by whitelisting rather than blacklisting. (Bug #25040566, Bug #22898475, Bug #80662)

• The performance of UTF-8 binary collations was improved. (Bug #24788778, Bug #83247, Bug #25076862)

• The systemd service file for mysqld now includes a Documentation value in the [Unit] section to provide a link to the systemd documentation in the MySQL Reference Manual. (Bug #24735762)

• Unit testing now uses Google Mock 1.8. (Bug #24572381, Bug #82823)

• The unimplemented and nonstandard WITH CUBE clause for GROUP BY is no longer supported.

  
  **Note**

  CUBE now is a reserved word and cannot be used as an identifier without identifier quoting.

  (Bug #24572048, Bug #82821)

• These outdated build scripts were removed from MySQL source distributions: BUILD/SETUP.sh, BUILD/autorun.sh, BUILD/cmake_configure.sh, BUILD/compile-dist, and cmake/configure.pl. (Bug #24512933)

• The removal of the mysql_shutdown() C API function in MySQL 8.0.0 has been reverted. It is still to be removed in a future MySQL version, but that removal has been deferred. (Bug #24496976, Bug #82681)

• Consistency and maintainability of Debian/Ubuntu packaging maintainer scripts was improved. (Bug #23588977)

• In-place ALTER TABLE operations are now possible for column modifications of tables containing generated columns that depend on columns with a DEFAULT value if the modified columns are not involved in the generated column expressions. For example, changing the NULL property of a separate column can be done in place without a table rebuild. (Bug #22987899, Bug #80832)

• For dumps of the mysql system database, mysqldump no longer generates DDL statements for the innodb_index_stats and innodb_table_stats tables. Such statements fail at dump reload time because those tables are not user accessible. (Bug #22655287)

• If mysqld is invoked with --daemonize, stdout and stderr are redirected to /dev/null if connected to a terminal type device, so that mysqld can behave as a true daemon. (Bug #21627629)

• mysql_secure_installation is more strict about what it considers valid yes and no responses. (Bug #13344753, Bug #62925)
MySQL 8.0 Release Notes

• MySQL client and utility programs reported their own program-specific version number when invoked
with the --version option, and the version strings did not follow a consistent format. Now these
programs print the MySQL version number (same as mysqld) in a consistent format. (Bug #11763201,
Bug #55885)
• A new mysqldump option, --network-timeout, enables large tables to be dumped by setting maxallowed-packet to its maximum value and network read and write timeouts to a large value. This
option is enabled by default. To disable it, use --skip-network-timeout. (Bug #11754493, Bug
#46103)
• Added two JSON aggregation functions JSON_ARRAYAGG() and JSON_OBJECTAGG(). The
JSON_ARRAYAGG() function takes a column or column expression as an argument, and aggregates the
result set as a single JSON array, as shown here:
mysql> SELECT col FROM t1;
+--------------------------------------+
| col
|
+--------------------------------------+
| {"key1": "value1", "key2": "value2"} |
| {"keyA": "valueA", "keyB": "valueB"} |
+--------------------------------------+
2 rows in set (0.00 sec)
mysql> SELECT JSON_ARRAYAGG(col) FROM t1;
+------------------------------------------------------------------------------+
| JSON_ARRAYAGG(col)
|
+------------------------------------------------------------------------------+
| [{"key1": "value1", "key2": "value2"}, {"keyA": "valueA", "keyB": "valueB"}] |
+------------------------------------------------------------------------------+
1 row in set (0.00 sec)

The order of the array elements is unspecified.
JSON_OBJECTAGG() takes two columns or expressions which it interprets as a key and a value,
respectively; it returns the result as a single JSON object, as shown here:
mysql> SELECT id, col FROM t1;
+------+--------------------------------------+
| id
| col
|
+------+--------------------------------------+
|
1 | {"key1": "value1", "key2": "value2"} |
|
2 | {"keyA": "valueA", "keyB": "valueB"} |
+------+--------------------------------------+
2 rows in set (0.00 sec)
mysql> SELECT JSON_OBJECTAGG(id, col) FROM t1;
+----------------------------------------------------------------------------------------+
| JSON_OBJECTAGG(id, col)
|
+----------------------------------------------------------------------------------------+
| {"1": {"key1": "value1", "key2": "value2"}, "2": {"keyA": "valueA", "keyB": "valueB"}} |
+----------------------------------------------------------------------------------------+
1 row in set (0.00 sec)

A NULL key causes an error; duplicate keys are ignored.
For more information, see Aggregate (GROUP BY) Functions. (Bug #78117, Bug #21647417)
• MySQL now supports an IMPORT TABLE statement that imports MyISAM tables based on information
contained in .SDI (serialized dictionary information) metadata files. This statement is useful for
exporting “raw” table files from one server and importing them into another, and provides a faster
215


alternative to dumping tables as a file of SQL statements using `mysqldump` and processing the dump file using `mysql`. For more information, see IMPORT TABLE Syntax.

- The deprecated `replace` utility has been removed. If you wish to continue using this utility, be sure to retain a copy from an installed version of MySQL before upgrading to the current version.

## Bugs Fixed

- **Incompatible Change**: The stored definition of a view for which an explicit column name list was provided could be invalid. For example, for this statement:

  ```sql
  CREATE VIEW v1 (name2) AS SELECT 1 AS name1 UNION SELECT 2 ORDER BY name1;
  ```

  The stored definition looked like this, which is invalid:

  ```sql
  CREATE VIEW v1 AS SELECT 1 AS name2 UNION SELECT 2 AS 2 ORDER BY name1;
  ```

  Now column names within the stored definition are not replaced by the column name list:

  ```sql
  CREATE VIEW v1 (name2) AS SELECT 1 AS name1 UNION SELECT 2 AS 2 ORDER BY name1;
  ```

  An incompatibility resulting from this change is that the `CREATE VIEW` statement can no longer be recreated solely from the `INFORMATION_SCHEMA VIEWS` table, because the `VIEW_DEFINITION` value does not show the column name list. Instead, you must also consult the `COLUMNS` table to get the column name list. Alternatively (and more simply), use `SHOW CREATE VIEW`. (Bug #23265335, Bug #81377)

- **InnoDB; Microsoft Windows**: On a MySQL 64-bit build on Windows, a file I/O retry result was misinterpreted due to a missing cast necessary for the correct operation of the retry path, resulting in a failing assertion and operating system error. (Bug #24711351)

- **InnoDB**: In debug builds, a call that initialized latch debugging raised an assertion due to a race condition. (Bug #25700405)

- **InnoDB**: The restriction that required the first undo tablespace to use space ID 1 was removed. The first undo tablespace may now be assigned a space ID other than 1. Space ID values for undo tablespaces are still assigned in a consecutive sequence. (Bug #25551311)

- **InnoDB**: A `DROP TABLE` operation raised an assertion on a server with an `innodb_force_recovery` setting of 5 or 6. `DROP TABLE` is no longer permitted with these `innodb_force_recovery` settings. (Bug #25385590)

  References: This issue is a regression of: Bug #19779113.

- **InnoDB**: Compiling the server without the Performance Schema caused a build failure. (Bug #25348787)

- **InnoDB**: During read-ahead, the wrong page size was used to calculate the tablespace size. (Bug #25330449)

- **InnoDB**: Compiling on Fedora 25 using `DWITH_LZ4=system` resulted in a build failure due to a deprecated `LZ4_COMPRESS_LIMITEDOUTPUT` function. (Bug #25297593)

- **InnoDB**: Disabling macros such as `UNIV_PFS_MUTEX`, `UNIV_PFS_RWLOCK`, and `UNIV_PFS_THREAD` caused compilation errors. (Bug #25251082)
MySQL 8.0 Release Notes

• **InnoDB:** A NULL virtual column field name in a virtual index caused a server exit during a field name comparison that occurs while populating virtual columns affected by a foreign key constraint. (Bug #25222337)

• **InnoDB:** The file handle type name for InnoDB file I/O Performance Schema instrumentation was changed from `os_pfs_file_t` to `pfs_os_file_t`. (Bug #25220118)

• **InnoDB:** Transactions with subqueries on `INFORMATION_SCHEMA` tables could block concurrent DDL operations. (Bug #25200952)

• **InnoDB:** During a range comparison, a secondary index field number was passed instead of clustered index field number, eventually causing the retrieval of an incorrect field. (Bug #25175249)

• **InnoDB:** A server exit on restart was caused by missing `my_thread_init()` and `my_thread_exit()` functions for background threads that initialize the `st_my_thread_var` structure. (Bug #25167032)

• **InnoDB:** A memcached read operation with a non-default read batch size configuration resulted in a server exit. (Bug #25147515)

• **InnoDB:** The `INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS` table reported `NULL` for a foreign key constraint name (`UNIQUE_CONSTRAINT_NAME`) after restarting the server. (Bug #25126722)

• **InnoDB:** A gap lock was taken unnecessarily during foreign key validation while using the `READ COMMITTED` isolation level. (Bug #25082593)

• **InnoDB:** Loading InnoDB tables required more memory in MySQL 5.7 due primarily to the addition of in-memory structure members introduced with temporary table optimizations. The in-memory structure members, only used for temporary tables, are now only allocated as needed. (Bug #25080442)

• **InnoDB:** After a `TRUNCATE TABLE` operation on a table with a `FULLTEXT` index, space size was incorrectly calculated resulting in an invalid read. (Bug #25053705)

• **InnoDB:** A prepared XA transaction was rolled back by a high priority transaction. The high priority transaction should wait if the blocking transaction is in a prepared state. (Bug #25032066)

• **InnoDB:** InnoDB passed an invalid argument to `syscall(SYS_futex)`. (Bug #24923840, Bug #83375)

• **InnoDB:** After redo log recovery, the node size of an undo tablespace object could sporadically be zero, eventually causing an error. Additionally, the undo tablespace object was incorrectly placed on the LRU list, and the header page of the undo tablespace could be present the buffer cache when the tablespace object is freed, causing a file-open failure on the first page read. (Bug #24916359)

• **InnoDB:** During a checkpoint, all `MLOG_FILE_NAME` redo log records were written in a single mini-transaction (mtr), causing a log parsing buffer overflow. (Bug #24793413, Bug #83245)

• **InnoDB:** The GCC `mach_parse_compressed` function should load one to five bytes depending on the value of the first byte. Due to a GCC bug, GCC 5 and 6 emit code to load four bytes before the first byte value is checked (GCC Bug #77673). A workaround prevents this behavior. Thanks to Laurynas Biveinis for the patch. (Bug #24707869, Bug #83073)

• **InnoDB:** Code related to tablespace type functions was improved to address issues with naming consistency, function distribution, and function usage. (Bug #24706739, Bug #83092)

• **InnoDB:** A mechanism was added to debug builds to ensure that keys for InnoDB Performance Schema instrumentation are registered with Performance Schema. The mechanism causes startup to fail on debug builds if the number of Performance Schema keys does not match the number of registered Performance Schema keys. (Bug #24686908)
- **InnoDB**: A race condition while updating table statistics could result in an estimated row count of 1 and an incorrect query execution plan. (Bug #24666839, Bug #82968)

- **InnoDB**: Due to a glibc bug, short-lived detached threads could exit before the caller had returned from pthread_create(), causing a server exit. Thanks to Laurynas Biveinis for the patch. (Bug #24605956, Bug #82886)

- **InnoDB**: An error in code related to table statistics raised an assertion in the `dict0stats.cc` source file. (Bug #24585978)

- **InnoDB**: The list of module base names representing modules in the InnoDB code base that allocate memory using `ut_malloc` or `ut_new` was incomplete. The list is used by the Performance Schema for reporting of InnoDB memory allocation event data. (Bug #24571816)

- **InnoDB**: Some InnoDB rw-lock and mutex keys were not registered for use with the Performance Schema. (Bug #24571597)

- **InnoDB**: After increasing the value of `innodb_undo_logs` and restarting the server, the number of active undo tablespaces was not increased when assigning undo tablespaces to newly allocated rollback segments. (Bug #24488141)

- **InnoDB**: The unused `MLOG_UNDO_HDR_DISCARD` redo log record type and related functions were removed. (Bug #24482001)

- **InnoDB**: An assertion was raised when the purge thread started due to the server not recognizing a tablespace ID as an undo tablespace ID. (Bug #24479773)

  References: This issue is a regression of: Bug #23517560.

- **InnoDB**: InnoDB incorrectly reported an error about missing encryption when restoring pages from the doublewrite buffer during recovery. (Bug #24471076)

- **InnoDB**: InnoDB reported an incorrect estimate for the number of pages in the buffer pool for a table. The number of pages was not decremented correctly when pages were removed. (Bug #24464147)

  References: This issue is a regression of: Bug #21747906.

- **InnoDB**: The following code changes related to undo tablespace management were implemented:
  
  - An in-memory array for temporary tablespace rollback segments was added, freeing 32 slots in the `TRX_SYS` page for use by redo-enabled rollback segments.

    As a result of this change, you no longer need to consider temporary tablespace rollback segments when setting the `innodb_rollback_segments` configuration option during configuration of separate undo tablespaces.

    - Code comments related to undo logs and rollback segments were revised.
    - Fixed arrays that list rollback segments and undo tablespaces were replaced by dynamic size arrays.
    - The function that initializes undo tablespaces was refactored.
    - A class was added for managing just-in-time creation of undo and undo file names.

    (Bug #24462978)

- **InnoDB**: A cached undo segment was not removed from the rollback segment history during a slow shutdown. (Bug #24450908)
• **InnoDB**: An error during a table-rebuilding operation on a table with only a generated clustered index (`GEN_CLUST_INDEX`) raised and assertion due to an error called with an invalid key name. (Bug #24444831)

• **InnoDB**: MySQL did not build with GCC 6.1.1. (Bug #24438752)

• **InnoDB**: Rotating the tablespace encryption master key while the server is in read-only mode raised an assertion instead of displaying an error message. (Bug #24404091)

• **InnoDB**: On a table without an explicitly defined primary key, InnoDB did not replace the implicit clustered index (`GEN_CLUST_INDEX`) when a unique key was defined on a NOT NULL column. (Bug #24397406)

• **InnoDB**: `handler::keys_to_use_for_scanning()` was removed from the handler API. The function was no longer used. The `HA_READ_ORDER` index flag provides the same information for each index. (Bug #24364448)

• **InnoDB**: Page cleaner threads asserted due to a regression related to the adaptive hash index feature. (Bug #24346574)

  References: This issue is a regression of: Bug #21407023.

• **InnoDB**: InnoDB failed to free memory used by the full-text optimizer thread. (Bug #24331265)

• **InnoDB**: During recovery, InnoDB attempted to fetch LOB pages using a null reference. (Bug #23615208)

• **InnoDB**: When adding a new index, the server dropped an internally defined foreign key index and attempted to use a secondary index defined on a virtual generated column as the foreign key index, causing a server exit. InnoDB now permits a foreign key constraint to reference a secondary index defined on a virtual generated column. (Bug #23533396)

• **InnoDB**: An INFORMATION_SCHEMA.FILES query resulted in a server exit due to a race condition with a concurrent tablespace creation operation. (Bug #23477214, Bug #81614)

• **InnoDB**: A concurrent DML operation during an in-place `ALTER TABLE` operation that rebuilt the table did not update a virtual index, resulting in a mismatch between the virtual index and clustered index. (Bug #23219499)

• **InnoDB**: A `TRUNCATE TABLE` operation held the `dict_sys` mutex while scanning for and removing pages from the buffer pool, causing concurrent DDL operations to stall. The mutex is now released during the scan and acquired again when the scan is completed. (Bug #23070734, Bug #80060)

• **InnoDB**: Inserting GIS data into an R-tree raised an assertion due to a missing page number field that was encountered when storing the B-tree cursor. (Bug #23044098, Bug #80939)

• **InnoDB**: Writing to an InnoDB internal temporary table did not increment the `Handler_write` counter. (Bug #23024178)

• **InnoDB**: Changes to the InnoDB recovery process in MySQL 5.7 could require up to three scans of the redo log during recovery. To reduce the number of scans, the first and second scans were merged. With this change, there is only one scan unless the redo log record hash table that is populated by the scan reaches its memory threshold. In this case, a second scan is initiated that performs a simultaneous scan and apply. (Bug #22963951, Bug #80788)

• **InnoDB**: A table-copying online `ALTER TABLE` operation on a `ROW_FORMAT=REDUNDANT` table with indexed virtual columns raised an assertion. (Bug #22018745)
• **InnoDB:** After a server restart, concurrent `INSERT` operations on a table with an auto-increment primary key resulted in a duplicate entry error. The current auto-increment value was not changed after `auto_increment_increment` and `auto_increment_offset` settings were modified. (Bug #20989615, Bug #76872)

• **InnoDB:** Performance Schema instrumentation for InnoDB I/O was disabled on Windows. (Bug #14025581)

• **InnoDB:** An automatically generated foreign key constraint identifier that exceeded the 64 character limit appeared truncated in the `INFORMATION_SCHEMA.TABLE_CONSTRAINTS CONSTRAINT_NAME` column. Automatically generated foreign key constraint identifiers can no longer exceed the 64 character limit. (Bug #11745347, Bug #13942)

• **InnoDB:** The `row_search_mvcc()` function unnecessarily traversed the entire table for a range query, which occurred when the record was not in the transaction read view. (Bug #84202, Bug #23481444, Bug #25251375)

• **Packaging:** The `my_create_minidump` function in the msys library called the Windows API function `MiniDumpWriteDump` with an incorrect value (NULL) for the `ExceptionParam` parameter. This resulted in exception information being omitted from the minidump file. (Bug #24505650, Bug #82695)

• **Partitioning:** For a partitioned table, dropping a partition dropped triggers for the table as well. (Bug #24449174)

• **Partitioning:** Updating a row of a table that had partitioning on a generated column could raise an assertion failure for debug builds, and return incorrect results in nondebug builds. (Bug #22574695, Bug #80080)

• **Replication:** MySQL 8.0.1 adds the original commit timestamp related statements, but `mysqlbinlog` was wrongly reporting that these statements have been present since MySQL 8.0.0 because they were written to the log with the incorrect version number 80000. The fix changes the version to 80001 in `mysqlbinlog` dumps. (Bug #25710507)

• **Replication:** MEMBER_STATE of a group replication member did not go from ERROR to OFFLINE when the `STOP GROUP_REPLICATION` command was executed if the error state was due to ER3092. (Bug #25674926)

• **Replication:** With flow control enabled, reaching a minimum flow control quota of 1 causes Group Replication to not stop throttling when the reason for throttling was no longer in effect. (Bug #25461354)

• **Replication:** Using an unresolvable host name in `group_replication_group_seeds` caused START GROUP_REPLICATION to fail. The fix ensures that host names in `group_replication_group_seeds` are validated when starting Group Replication and the list must contain at least one valid address. Invalid addresses are ignored. (Bug #25460324, Bug #84674)

• **Replication:** Lock contention impeded binary relay performance during processing of the relay log file on the slave. (Bug #25321231, Bug #77778)

• **Replication:** A failed node that was removed from the cluster could be expelled upon rejoining the cluster. (Bug #25311008)

• **Replication:** The `_gr_user` account created by Group Replication plugin installation was not reliably removed when the plugin was uninstalled. (Bug #25298987)

• **Replication:** When starting Group Replication on an offline node, the node could be configured for replication, but fail for recovery. (Bug #25256910)

• **Replication:** When using a multithreaded slave, applier errors displayed worker ID data that was inconsistent with data externalized in Performance Schema replication tables. (Bug #25231367)
MySQL 8.0 Release Notes

- **Replication:** Not all Group Replication GCS debug and trace messages were enabled in debug mode. (Bug #25209109, Bug #84079)

- **Replication:** Compiling MySQL 5.7.17 failed with a variable length array error. (Bug #25163241, Bug #83994)

- **Replication:** In row-based replication, a message that incorrectly displayed field lengths was returned when replicating from a table with a `utf8mb3` column to a table of the same definition where the column was defined with a `utf8mb4` character set. (Bug #25135304, Bug #83918)

- **Replication:** Group Replication GCS was not discarding messages when a member within the group was inactive. (Bug #25134074)

- **Replication:** Some unnecessary warnings were given when the Group Replication plugin was compiled on Windows platforms. (Bug #25119288)

- **Replication:** If the binary log on a master server was rotated and a full disk condition occurred on the partition where the binary log file was being stored, the server could stop unexpectedly. The fix adds a check for the existence of the binary log when the dump thread switches to next binary log file. If the binary log is disabled, all binary logs up to the current active log are transmitted to slave and an error is returned to the receiver thread. (Bug #25076007)

- **Replication:** As assertion could be raised if the Group Replication plugin attempted to contact the server when that was no longer possible. (Bug #25071492)

- **Replication:** The GTID transaction skipping mechanism that silently skips a GTID transaction that was previously executed did not work properly for XA transactions. (Bug #25041920)

- **Replication:** After executing restarts on the group replication applier SQL thread, the plugin could no longer detect failure of the thread. (Bug #24969065)

- **Replication:** Building Group Replication on Windows requires a minimum CMAKE version of 2.8.12. (Bug #24964522)

- **Replication:** If a relay log index file named relay log files that did not exist, `RESET SLAVE ALL` sometimes did not fully clean up properly. (Bug #24901077)

- **Replication:** When the MTS slave applier stopped because of an (injected) error, it reported no useful information for troubleshooting. (Bug #24822686)

- **Replication:** `FLUSH BINARY LOGS` could become slow with data replicated from many servers. (Bug #24806259, Bug #83270)

- **Replication:** When using XA transactions, if a lock wait timeout or deadlock occurred for the applier (SQL) thread on a replication slave, the automatic retry did not work. The cause was that while the SQL thread would do a rollback, it would not roll the XA transaction back. This meant that when the transaction was retried, the first event was `XA START` which was invalid as the XA transaction was already in progress, leading to an `XAER_RMFAIL` error. (Bug #24764800)

  References: See also: Bug #83588, Bug #24923091, Bug #24966941.

- **Replication:** Enabling the group replication plugin caused the `performance_schema_max_mutex_classes` default value of 200 to be exceeded. As a result, some group replication mutex instruments did not appear in the `performance_schema.setup_instruments` table. (Bug #24746530)

- **Replication:** A partially failed `CREATE USER`, `RENAME USER`, or `ALTER USER` statement was not correctly consuming an auto-generated or specified GTID when binary logging was disabled. (Bug #24693798)
- **Replication:** Binlog_sender, which writes events from the binary log to a packet buffer and then sends the packet to the slave, did not reduce the size of the send buffer as expected. (Bug #24643036)

- **Replication:** The group commit update of GTIDs has been refactored to improve performance on workloads with many small transactions. (Bug #24398760)

- **Replication:** If the relay_log option was not specified in a configuration file, the relay_log_basename variable was being internally constructed on the fly using hostname but the relay_log_basename variable was not set. When a slave tried to access this uninitialized variable it resulted in an unexpected halt of the server. (Bug #24352667)

- **Replication:** For servers built with yaSSL, using group replication with encrypted connections could result in timeout failures waiting for view delivery. (Bug #23592214)

- **Replication:** When using a multithreaded slave (slave_parallel_workers greater than 0) the value of Seconds_Behind_Master was incorrect when rotating a relay log. (Bug #23532304)

- **Replication:** An XA PREPARE statement that failed during the intermediate steps could lead to an inconsistent XA transaction state, where ID = -1 but the binlogged flag was set to true. This caused asserts while executing XA COMMIT and XA ROLLBACK queries. (Bug #22915670)

- **Replication:** The server prevented several replication-related administrative statements from working if the read_only system variable was enabled. (Bug #22857926, Bug #25363745, Bug #25326058, Bug #84350, Bug #84437)

- **Replication:** CHANGE MASTER TO for a channel that did not exist could raise an assertion. (Bug #22255698)

- **Replication:** The delay specified by the binlog_group_commit_sync_delay system variable was applied to too many binary log commit groups. (Bug #21420180)

- **Replication:** The number of generated unwanted fseeks into the binary log file being replicated to a slave has been reduced. (Bug #83226, Bug #24763579)

- **Replication:** The fix for Bug #81657 was not correctly merged into MySQL 8.0. Thanks to Laurynas Biveinis for alerting us. (Bug #83124, Bug #24715790)

- **Replication:** The rpl.rpl_binlog_errors test was failing sporadically on Windows. (Bug #82302, Bug #24330138)

- **Replication:** When binlog_group_commit_sync_delay was set to a value between 1 and 9, if binlog_group_commit_sync_no_delay_count was set to a value greater than 1, and the number of transaction commits was less than binlog_group_commit_sync_no_delay_count, these commits hung forever if no more commits were received; and if binlog_group_commit_sync_no_delay_count was set to 0, all transaction commits hung forever. (Bug #80652, Bug #22891628)

- **Replication:** Concurrent CREATE TRIGGER and DROP TRIGGER statements were not being binary logged in the correct order, causing slaves to fail. (Bug #77095, Bug #21114768)

- **Microsoft Windows:** On Windows, SHOW TABLES FROM db_name hung if db_name was given in uppercase. (Bug #24800048, Bug #83262)

- **Microsoft Windows:** 32-bit builds are no longer supported on Windows, but CMake failed to detect when a 32-built build would be attempted, resulting in compilation errors later. Now CMake detects 32-bit build attempts and produces an appropriate error message. (Bug #24487483, Bug #82645)

- **Solaris:** When the WITH_INNODB_MEMCACHED CMake option is enabled, memcached now can be built on Solaris 11 and 12 using the Developer Studio 12.5 compiler. (Bug #24504155, Bug #82692)
MySQL 8.0 Release Notes

- **Solaris:** Library search path handling on Solaris was incorrect. (Bug #24487934, Bug #82646)

- **JSON:** When a JSON value consisted of a large sub-document wrapped in many levels of JSON arrays, objects, or both, serialization of the JSON value sometimes required an excessive amount of time to complete. (Bug #23031146)

- **JSON:** When a NULL value existed in a JSON column, the result from a query using `GROUP_CONCAT()` together with the `ORDER BY` clause was not always correct. (Bug #22992666)

- **JSON:** The internal `rapid_json_handler` used its own data structures to represent a partially-built DOM; these had to be converted into a `Json_dom` graph before returning the result. Now this handler builds the graph directly, which reduces the amount of work required to build it, and thus to parse a JSON document. (Bug #22900110)

- **JSON:** The internal function `Item_func_case::val_json()` did not always set the null value flag as expected when a CASE expression evaluated to NULL, leading to an assertion in debug builds of the server. (Bug #22887227)

- **JSON:** The `SUM()` function truncated decimal values extracted from JSON documents, producing an integer result. (Bug #84935, Bug #25530204)

- **JSON:** A JSON document that contained a double value slightly greater than the maximum value that can be represented by a double silently replaced it with zero instead of rejecting the value and raising an error. Such values are now handled correctly in MySQL JSON documents.

  The underlying issue was traced to a problem with RapidJSON, which has been reported to that library’s developers as Issue #849. (Bug #84891, Bug #25518504)

- **JSON:** The `JSON_SEARCH()` and `JSON_CONTAINS_PATH()` functions did not work when the `one_or_all` argument was specified using UTF-16 encoding. For both of these functions, this argument is now converted to `utfmb4` if need be before its value is checked. (Bug #84880, Bug #22516960)

- **JSON:** The `JSON_UNQUOTE()` function did not work with strings that used UTF-16 encoding. Now these strings are converted to `utfmb4` internally before being processed. (Bug #84878, Bug #25516881)

- **JSON:** Updating the same JSON column in a single statement could cause incorrect values to be written into the table. This occurred when the second update overwrote the column value with a subset of itself. An example of such a statement is shown here:

  ```sql
  UPDATE t SET col = JSON_ARRAY(value), col = col->'$[0]';
  ```

  (Bug #84694, Bug #25461627)

- **JSON:** The functions `JSON_QUOTE()` and `JSON_UNQUOTE()` did not work correctly with multibyte character sets such as `utf8mb4`. (Bug #84680, Bug #25455065)

  References: See also: Bug #77234, Bug #21193273.

- **JSON:** The error message for Error 3152 `ER_JSON_USED_AS_KEY` has been changed from JSON column '%s' cannot be used in key specification to the less confusing and more accurate JSON column '%s' supports indexing only via generated columns on a specified JSON path. (Bug #81364, Bug #23274244)

- **JSON:** Internal tests for MySQL JSON functionality ran out of stack space on some platforms when run against a debug-enabled server. Because timely checks were not made for stack usage, the server did not detect this situation, leading to a server exit.
The fix for this issue is twofold:

- Stack overrun checks are now made before attempting to serialize a nested array or object, so that the operation fails gracefully when processing deeply nested JSON documents, rather than causing an exit.

- Serialization of JSON documents has been reorganized so that it requires less use of the stack when compiled without optimization.

(Bug #81083, Bug #23106330, Bug #26399306)

- BIN() could produce incorrect truncation with multibyte character sets when used within UNION. (Bug #27511490, Bug #89581)

- MySQL did not compile with GCC 7. (Bug #25643811, Bug #26825211)

- The (undocumented) WINDOWS_RUNTIME_MD CMake option has been removed. (Bug #25611359)

- If --skip-innodb or one of its variants was used, a spurious warning about avoid_temporal_upgrade was generated. (Bug #25573578)

- mysqld_safe failed to restart the server if a PID_FILE.shutdown file was present. (Bug #25572504)

References: This issue is a regression of: Bug #11751149.

- For Debian/Ubuntu packages, user-defined collation files could be overwritten during MySQL upgrades.Charset files are now marked as conffiles so that user customizations generate a prompt during upgrades whether to overwrite them. (Bug #25525628, Bug #84761)

- For CREATE TABLE statements that specified the table name with a database qualifier and included a DATA DIRECTORY or INDEX DIRECTORY option, an error occurred if there was no default database. (Bug #25514146, Bug #84861)

- referenced_table_schema and referenced_table_name field values in the mysql.foreign_keys data dictionary table were not stored in lowercase when lower_case_table_names was enabled. (Bug #25495714)

- Starting the server with performance_schema_digests_size=1 caused an abnormal exit. (Bug #25492129, Bug #84786)

- For clients linked against yaSSL, connections became invalid when a read timeout occurred, rather than retrying the read. (Bug #25444075)

- MySQL compilation in different directories produced different builds to leakage of absolute paths into debug information and __FILE__. (Bug #25436469, Bug #84608, Bug #25859274, Bug #85855)

- Calculations for UCA 9.0.0 collations were inefficient for tailoring rules containing contraction characters. (Bug #25426632, Bug #84577, Bug #25426632, Bug #84577)

- A negative internal connection timeout value caused the connection to go idle and abort. A negative value now causes the connection to block indefinitely in the absence of I/O. (Bug #25408557)

- A mutex Performance Schema name was too long and produced a warning at server startup. (Bug #25406915)

- MySQL failed to compile on some platforms with -DWITH_LIBWRAP=ON. CMake support now checks whether tcpd.h has proper function prototypes. (Bug #25395543, Bug #84495)
• An index defined on a data dictionary table column exceeded the maximum index key length when the instance was initiated with `innodb_page_size=4k`. (Bug #25384527)

• `mysqld_safe` did not check whether the directory named by the `--basedir` option existed. (Bug #25365194)

• Configuring `CMake` with `-G ninja` resulted in build output that was inappropriate for build platforms other than Xcode or Visual Studio. (Bug #25358460)

• `mysqld_safe` failed if the error log file named by the `--log-error` option was a FIFO. (Bug #25356221, Bug #84427)

• For prepared statements, an alias within a subquery or derived table might cause incorrect behavior during statement execution if another alias depended on it. (Bug #25343335, Bug #84398, Bug #25171608)

• `mysqld_safe` could fail if the `--datadir` option value ended with a `/` character. (Bug #25319457)

• A recent change to `mysqld_safe` caused the `mysql.server` script to be unable to start it if the base directory was specified as an absolute path that differed from the compiled-in default absolute path. (Bug #25319392, Bug #84263)

• The `CONNECTION_CONTROL` plugin failed to compile if the Performance Schema was disabled. (Bug #25308357, Bug #84304)

• Passwords did not expire correctly for accounts created using MySQL Workbench. (Bug #25299309)

• For System V init scripts for RPMs, the `[mysqld]` option-file section was being ignored for some options, such as `pid-file`. (Bug #25287707, Bug #84172)

• Init scripts failed to launch `mysqld_safe` if a non-default base directory was used. (Bug #25261472, Bug #84219)

• `CMake` now detects whether a GCC 5.3.0 loop optimization bug occurs and attempts a workaround if so. (Bug #25253540)

• `mysqld_safe --no-defaults` did not work (inadvertent consequence of an earlier bug fix). (Bug #25244898, Bug #84173)

• Semicolon (`;`) characters within or between statements could cause distinct digests to be generated from identical statements. (Bug #25244533, Bug #83253)

• Components could not register services without referring to their private implementation. A `SERVICE_IMPLEMENTATION(component, service)` macro now enables this to be avoided. (Bug #25238906)

• Certain stored functions, if used in a query `WHERE` clause, could be handled using Index Condition Pushdown (which should not happen), resulting in a server exit. (Bug #25196653, Bug #25174454)

• For `ai_ci` collations based on Unicode Collation Algorithm 9.0.0, accented characters that compare equal were treated as different by `LIKE` comparisons. (Bug #25167284, Bug #83999)

• For a client linked against `libmysqlclient`, invalid memory access could occur during use of prepared statements. (Bug #25164932)

• Some Linux startup scripts did not process the `datadir` setting correctly. (Bug #25159791)

• If a character set is specified for a column of a partitioned table, a segmentation fault could occur while upgrading. (Bug #25153261)
MySQL 8.0 Release Notes

• **LOAD DATA** failed to accept multibyte characters that followed an escape sequence. (Bug #25147988, Bug #83950, Bug #25865525)

• The fix for Bug #25088048 caused the command used by `mysqld_safe` to start the MySQL server to no longer include the `mysqld` path. (Bug #25144379)

  References: This issue is a regression of: Bug #25088048.

• For UCA collations, **LIKE** comparisons against a pattern that ended with the escape character returned incorrect results. (Bug #25140629, Bug #83930)

• The default character set and collation were used instead of the character set and collation defined in `db.opt` file when upgrading the schema from MySQL 5.7 to MySQL 8.0. (Bug #25139901)

• Instead of updating the data dictionary cache at the end of DDL statements using separate function calls, the data dictionary cache is now updated as part of transaction commit. (Bug #25095798, Bug #83818)

• **CREATE TABLE** now requires the **FILE** privilege if **DATA DIRECTORY** or **INDEX DIRECTORY** is specified explicitly as a table or partition option. **ALTER TABLE** requires the **FILE** privilege if either option is specified explicitly as a partition option (it ignores them if specified as table options). (Bug #25092566)

• There were some differences between ICU order and MySQL Unicode collations for Hungarian contractions and ligatures. MySQL now follows ICU order. (Bug #25090543)

• The data dictionary failed to initialize and start when using a binary collation. The query string generated to create the data dictionary schema did not add quotes to the binary collation name. (Bug #25054104, Bug #83706)

• Executing a stored procedure containing a query that accessed a view could allocate memory that was not freed until the session ended. (Bug #25053286)

• Compilation on FreeBSD 11 failed attempting to check **MAP_NORESERVE**, which is no longer defined. (Bug #25048128, Bug #83689)

• `mysql-test-run.pl` now checks whether the **TSAN_OPTIONS** environment variable is set. If so, the value is taken as the path name of a file containing ThreadSanitizer suppressions (errors to be ignored during test runs). Additionally, the **--sanitize** option now causes `mysql-test-run.pl` to scan the server error logs for ThreadSanitizer messages. (Bug #24970905, Bug #83601)

• After starting the server with a nonzero **--lower-case-table-names** setting, an assertion was raised when **USE INFORMATION_SCHEMA**; was the first instruction from the first client. (Bug #24963580)

• For case-insensitive Unicode collations, the various space characters did not hash to the same value, resulting in incorrect comparisons between them. (Bug #24956750, Bug #83549)

• Dictionary clients now track uncommitted dictionary objects that are being modified by DDL statements, making uncommitted changes by the same session visible to a dictionary client without affecting the dictionary object cache prior to commit time. (Bug #24956365, Bug #83548)

• Simultaneous dictionary object cache misses were not handled correctly. An object retrieved by one thread could be evicted from the cache before a waiting thread attempted to access it. (Bug #24949179)

• **CMake** now avoids configuring the **-fexpensive-optimizations** option for GCC versions for which the option triggers faulty shift-or optimizations. (Bug #24947597, Bug #83517)

• **CMake** support was added for compiling using the **-std=c++03** option under Developer Studio 12.5. This is now used rather than **stlport** by default. (Bug #24947136, Bug #83512, Bug #25229424)
MySQL 8.0 Release Notes

- **NCHAR** and **NATIONAL CHAR** are synonyms, but `CAST(expr AS NCHAR)` succeeded, whereas `CAST(expr AS NATIONAL CHAR)` did not. Now both work. (Bug #24934161)

- OEL RPM packages now better detect which platforms have multilib support (for which 32-bit and 64-bit libraries can be installed). Thanks to Alexey Kopytov for the patch. (Bug #24925181, Bug #83457)

- OEL RPM packages now better detect which platforms do not have multilib support (for which 32-bit and 64-bit libraries can be installed). Thanks to Alexey Kopytov for the patch. (Bug #24916428, Bug #83428)

- The `LOCATE()` function returned `NULL` if the `substr` or `str` argument was `NULL`, but not if the `pos` argument was `NULL`. Now it returns `NULL` if any argument is `NULL`. (Bug #24911350, Bug #83427)

- Bit operations could cause a server exit to occur if argument nullability was mishandled. (Bug #24910958, Bug #24930829)

- The Block Nested Loop algorithm could allocate too much memory during query execution. (Bug #24909223)

- Information about building MySQL 5.6 compatibility libraries in the MySQL 5.7 and higher `.spec` file is needed only for building `libmysqlclient` and `libmysqld`. Information about building the InnoDB memcached plugin was removed. (Bug #24908345, Bug #83409)

- Incorrect updating of view metadata could raise an assertion. (Bug #24834622)

- Initialization of the `keyring_okv` plugin failed if the `STANDBY_SERVER` setting was missing from the `okvclient.ora` configuration file, effectively making this a mandatory setting. `STANDBY_SERVER` is now optional. (Bug #24816271)

- Data dictionary objects acquired by `Dictionary_client::acquire_uncached()` are now owned by the current auto-releaser instead of the caller. Also, acquisition of `TABLE_SHARE` view objects is performed by `acquire()` and `clone()` instead of `acquire_uncached()`. (Bug #24813358, Bug #83296)

- Privilege checking could be incorrect for a derived table used within a multiple-table `UPDATE` invoked within a stored procedure or view object, for the second or subsequent execution of the object, if the derived table was merged into the outer query. (Bug #24810564)

- The `Created_tmp_tables` status variable was incremented in some cases when no temporary table was created. (Bug #24808970, Bug #83287)

- An in-place upgrade from MySQL 5.7 to MySQL 8.0 failed if parsing of a stored routine body failed while migrating the routine. Now a warning is reported and the routine is created without parsing its body. Also, warnings are now reported when dependency resolution fails for a view during view migration. (Bug #24805140, Bug #83275)

- For compilation, `mysql_upgrade` is dependent on the dynamically generated `sql_commands_system_tables_data_fix.h` file, but a missing dependency could cause that file not to be generated. (Bug #24802377, Bug #83272)

- A race condition between transactions accessing the access-control list (ACL) cache and use of the ACL cache to populate `INFORMATION_SCHEMA` tables could cause a server exit. (Bug #24786029)

- For debug builds, `EXPLAIN` or `DESCRIBE` for a table with a database or table name longer than the maximum permitted length raised an assertion rather than displaying an appropriate error. (Bug #24751177, Bug #83114)

- The implementation of several `INFORMATION_SCHEMA` tables as views on data dictionary tables introduced a number of native SQL functions intended only for internal use by the server, but they could
be invoked by users. Those functions now produce an error if invoked by users. (Bug #24749248, Bug #83189)

• CASE, COALESCE(), IF() and IFNULL() could merge a mix of signed and unsigned arguments incorrectly and produce an incorrect result type. (Bug #24733658, Bug #83148)

• Connections from a client to a server with SSL enabled succeeded even if --ssl-mode had a value of VERIFY_CA or VERIFY.IDENTITY and the client did not provide a CA certificate. (Bug #24732452, Bug #23189252, Bug #25397416, Bug #84508)

• Manual creation of a directory in the data directory resulted in USE dir_name succeeding even though the directory was not registered as a database in the data dictionary. (Bug #24732194, Bug #83140)

• If InnoDB statistics were incorrect, FOUND_ROWS() could return 1 even when the previous SELECT returned no rows. (Bug #24714857, Bug #83110)

• ALTER TABLE on a temporary table could raise an assertion if a nontemporary table with the same name existed. (Bug #24713918, Bug #83117)

• CMake now sets -DWITH_NUMA=ON for Debian platforms where possible. (Bug #24689101)

• To better provide atomic file creation, Debian packaging scripts now use the coreutils install command rather than touch, chmod, and chown. (Bug #24688682)

• Enabling the DISABLE_SHARED CMake option caused compilation failure. (Bug #24687701, Bug #83039)

References: This issue is a regression of: Bug #24481181.

• The BIN(), OCT(), and HEX() functions could mishandle values of the BIT data type. (Bug #24686658, Bug #83031)

• For debug builds, an invalid utf8 character in the comment of a CREATE EVENT or ALTER EVENT statement raised an assertion. This now produces an ER_INVALID_CHARACTER_STRING error. (Bug #24679962)

• The encoding for the utf8 character set permitted characters between U+D800 and U+DFFF as valid, though they are reserved for surrogate pairs and do not directly represent characters. They are now considered invalid. (Bug #24672415)

• A query could produce incorrect results if the WHERE clause contained a dependent subquery, the table had a secondary index on the columns in the select list followed by the columns in the subquery, and GROUP BY or DISTINCT permitted the query to use a Loose Index Scan. (Bug #24671968, Bug #83005)

• After a RENAME TABLE operation on a table with a foreign key, the generated constraint name was not updated in TABLE_CONSTRAINTS. (Bug #24666169)

• On macOS, CMake AddressSanitizer support did not work. (Bug #24661626, Bug #82976)

References: This issue is a regression of: Bug #23759968.

• Compilation failed on macOS 10.11 with Xcode 8.0. (Bug #24661523, Bug #82975)

• Index hints applied to invisible indexes produced no error. (Bug #24660093, Bug #82960)

• In some cases, INFORMATION_SCHEMA.KEY_COLUMN_USAGE did not report data for all foreign key constraints. (Bug #24655803, Bug #82961)

• REPLACE() on large strings could be slow and unkillable. (Bug #24652792)
• The `DebugPrintTest` and `DebugPrintDeathTest` unit tests did not handle divide-by-zero testing properly on the Aarch64 platform. Thanks to Alexey Kopytov for the patch. (Bug #24624555, Bug #82889)

• Compilation on FreeBSD using GCC 6 did not work. (Bug #24619561, Bug #82922)

• Changes made to `mysqld_safe` in recent MySQL releases require the `--ledir`, `--mysqld`, `--mysqld-version` options to be specified on the command line; they can no longer be specified in option files. This could cause failure of init scripts that invoke `mysqld_safe`. Such scripts now pass the value of the `MYSQLD_OPTS` environment variable as the first command-line argument to `mysqld_safe`, with the value set to such command line-only `mysqld_safe` option values as may be required. On platforms that use systemd, the `MYSQLD_OPTS` value can be set in `/etc/sysconfig/mysqld` with a line such as this:

```
MYSQLD_OPTS=" --ledir=/mysqld_ledir --mysqld=my_wrapper "
```

The value of `MYSQLD_OPTS` can also include `mysqld` options for `mysqld_safe` to pass to `mysqld`. (Bug #24619033, Bug #82920)

References: This issue is a regression of: Bug #24464380, Bug #24483092, Bug #25088048, Bug #25378439, Bug #25378565.

• SET PERSIST `innodb_buffer_pool_size = value` stored the original value of `innodb_buffer_pool_size` to `mysqld-auto.cnf`, not the new value. (Bug #24613005, Bug #82905)

• For SLES packages, a typo in the installation script postamble prevented some cleanup from occurring. (Bug #24605300, Bug #82389)

• Some messages written by the server to the error log while upgrading the data directory were missing the standard timestamp and process ID information. (Bug #24600054, Bug #82874)

• It was possible for the session values of the `Last_query_cost` and `Last_query_partial_plans` status variables to be accessed before they had been initialized. (Bug #24596263)

• Warnings occurring during `CREATE TABLE ... SELECT` could cause a server exit. (Bug #24595992)

• For `LOAD DATA` used to insert data into an updatable view, the check to verify whether a column is actually updatable was missing. (Bug #24595937)

• The server could dereference a null pointer when a deterministic function returning `LONGTEXT` was used in a subquery. (Bug #24595581)

• A view altered with `ALTER VIEW` might be dropped if the statement failed with an error. (Bug #24594140)

• Conversion of `JSON` documents to string could be slow if the document was large and contained many signed integers. (Bug #24586888)

• The data dictionary can contain entries for temporary tables (names beginning with `#sql`). These tables were exposed to `INFORMATION_SCHEMA` queries and through `SHOW` statements. This could cause `mysqldump` and `mysqlpump` to fail when they attempted to dump such tables. Temporary tables are now hidden to `INFORMATION_SCHEMA` queries and `SHOW` statements. (Bug #24580599, Bug #24571427)

• When a view was defined, the character set of the definition was not considered. If this differed from the default character set, table names used in the view might not be recognized, causing the view to be marked invalid.
For debug builds, an assertion could be raised if an error occurred while parsing an `ALTER VIEW` statement and the diagnostics area was empty. (Bug #24580586)

- For `LOAD DATA` statements, input data with too many column values produced only a warning, rather than an error as in MySQL 5.6. An error now occurs. (Bug #24577194, Bug #82830)

- Using `SET PERSIST` with the `global_log` system variable was ineffective. (Bug #24569624, Bug #82807)

- In the Performance Schema `variables_info` table, the `VARIABLE_SOURCE` column was not set properly for some boolean options specified in option files without a variable. (Bug #24567960)

- The `.mylogin.cnf` option file is intended for use by client programs, but the server was reading it as well. The server no longer reads it. (Bug #24557925)

- Use of `boost::chrono` was replaced with `std::chrono` because the former was causing link errors. (Bug #24556808, Bug #82781)

- The X Plugin was built with compilation options different from other plugins. (Bug #24555770, Bug #82777)

- When populating the `variables_by_thread` table, the Performance Schema could attempt to access session variables of other threads that were being deinitialized. (Bug #24555658)

- Users without proper privileges could load and unload server components. (Bug #24528148)

- Concurrent execution of `INSTALL COMPONENT` and `UNINSTALL COMPONENT` statements could cause a server exit. (Bug #24527148)

- On Debian/Ubuntu platforms, the systemd startup script for MySQL ignored `datadir` settings in `/etc/mysql/my.cnf`. (Bug #24517024, Bug #82709)

- With a `LOCK TABLES` statement in effect, DML statements on a table that had triggers could cause a server exit. (Bug #24506766)

- Parallel slave threads running account-management statements could fail due to a race condition in handling privilege cache locks. (Bug #24503606)

- Executing `ALTER TABLE` on a table that has triggers concurrently with other DDL operations could result in a corrupted data dictionary. (Bug #24497803)

- If `mysqladmin shutdown` encountered an error determining the server process ID file, it displayed an error message that did not clearly indicate the error was nonfatal. It now indicates that execution continues. (Bug #24493829, Bug #82670)

- For the `null_audit` plugin, setting the `null_audit_event_record` system variable improperly could cause a server exit. This variable should be set only from within the `null_audit` plugin, so it is now read only. (Bug #24493829, Bug #82670)

- In certain contexts, an expression such as `IF(col_name > 5000, (1 / col_name), 5000)` could get a type of `DECIMAL(6,4)`, which would truncate results. (Bug #24492965, Bug #82668)

- The data structure used for `ZERO_FILL` columns could experience memory corruption, leading eventually to a server exit. (Bug #24489302)

- Operation of the `mysql-multi.server.sh` script was based on `my.cnf` in the data directory. That option file is no longer used, so `mysql-multi.server.sh` has been removed. (Bug #24487870)
• A query could produce incorrect results if \texttt{MIN()} or \texttt{MAX()} in a subquery referred to an indexed column. (Bug #24484060, Bug #82638, Bug #24657798, Bug #82965)

• \texttt{SHOW TRIGGERS} output order could differ on Linux and Windows. (Bug #24482919, Bug #82637)

• \texttt{mysqld_safe} attempted to read \texttt{my.cnf} in the data directory, although that is no longer a standard option file location. (Bug #24482156)

• After running \texttt{mysql_upgrade}, executing an \texttt{INSTALL COMPONENT} statement could cause a server exit. (Bug #24453571)

• A regular expression pattern match into a large string could result in a server exit due to memory allocation failure or integer overflow. (Bug #24449076, Bug #24449090)

• Starting the server with a MySQL 5.7 data directory resulted in failure due to absence of the \texttt{default_roles} and \texttt{role_edges} system tables. (Bug #24447771)

• An incorrect error was reported for \texttt{CREATE TABLE} statements with a large value for the \texttt{CONNECTION} table option. The value is now limited to 1024 bytes. (Bug #24437124)

• An assertion could be raised if an \texttt{ER_LOCK_WAIT_TIMEOUT} error occurred during execution of \texttt{SHOW CREATE TRIGGER}. (Bug #24420809, Bug #82483)

• Constant folding could produce incorrect results for large unsigned integers. (Bug #24401273, Bug #82425)

• Use of very long subpartition names could result in a server exit. Now partition or subpartition names larger than 64 characters produce an \texttt{ER_TOO_LONG_IDENT} error. (Bug #24400628, Bug #82429)

• The \texttt{Gis_wkb_vector<Gis_point>} copy constructor was not explicitly instantiated, causing build problems for the Intel compiler. (Bug #24397833, Bug #82358)

• Privilege escalation was possible by exploiting the way \texttt{REPAIR TABLE} used temporary files. (Bug #24337928)

• A race condition between \texttt{UNINSTALL PLUGIN} and \texttt{SHOW PLUGINS} could result in a server exit. (Bug #24344026)

• With \texttt{ROW} mode binary logging, component installation raised an assertion. (Bug #24343582)

• A potential memory leak related to roles was fixed. (Bug #24337928)

• Subqueries that were converted to semi-joins and programmatically generated an \texttt{INFORMATION_SCHEMA} table could incorrectly treat the \texttt{INFORMATION_SCHEMA} table as empty.

  A workaround for this problem prior to the bug fix: \texttt{SET optimizer_switch='semijoin=off';} (Bug #24287772, Bug #82214)

• \texttt{mysqldumpslow} failed to parse timestamps in the slow query log; it had not been updated to track a change in log timestamp format. (Bug #24007040)

• The AppArmor profile installed by Ubuntu packages was missing an entry permitting \texttt{libnuma} to read a /\texttt{sys} hierarchy path, resulting in server startup failure. (Bug #23854929)

• The optimizer could incorrectly treat \texttt{RAND()} as a constant for queries subjected to semi-join transformation. (Bug #23854015)

• For debug builds, unequal-length binary operands for bit operators could raise an assertion. (Bug #23853628)
If a transaction rollback request rolled back only a statement and not the whole transaction, any attempt at attaching another operation to the transaction raised an assertion. (Bug #23753319, Bug #82143)

STRCMP() for arguments with the utf8mb4_unicode_ci collation could return results different from ORDER BY STRCMP() has been corrected. (Bug #23752284, Bug #82132)

Complete logical backups made with mysqlpump could not be restored if GTIDs were enabled. To enable control over GTID information written to the dump file, mysqlpump now has a --set-gtid-purged option that indicates whether to add a SET @@GLOBAL.gtid_purged statement to the output. (Bug #23748432)

Infinite recursion could occur if the audit_log plugin signalled an error while handling an error. (Bug #23717558, Bug #82052)

Messages written by the audit_log plugin to the error log regarding MYSQL_AUDIT_CONNECT event failures now print the underlying error cause as well to aid debugging. (Bug #23710632)

MySQL now uses readdir() rather than readdir_r(). The latter has been deprecated since glibc 2.24 and caused debug builds of MySQL and builds using GCC 6.1 to fail. Additionally, several problems resulting in GCC 6.1 compiler warnings were corrected. (Bug #23708395, Bug #24437737, Bug #82515, Bug #24459890, Bug #82583, Bug #25103242)

FORCE INDEX was ineffective for SELECT COUNT(*) queries. (Bug #23596760, Bug #81854)

During startup, the server creates a lock file for the Unix socket file (for example, mysql.sock.lock as a lock file for mysql.sock). If the server failed to write the process ID to the lock file, it failed to remove that file, which could cause subsequent server startups to fail until the file was removed manually. (Bug #23582603, Bug #81838)

For debug builds, queries executed using Item_func_spatial_collection::val_str() could raise an assertion. (Bug #23573720)

For audit log events in the connection class, the connection_type value was available only for connect events. The value is now available in connect, disconnect, and change-user events. (Bug #23541550)

The audit_log plugin audit_log_filter_remove_filter() function caused a server exit if given a NULL argument. (Bug #23522793)

Attempts to configure MySQL without the FEDERATED storage engine failed (the CMake -DWITH_FEDERATED_STORAGE_ENGINE=0 option did not work). (Bug #23508203, Bug #81665)

On Solaris, gettimeofday() could return an invalid value and cause a server shutdown. (Bug #23499695)

The keyring_file plugin could attempt to write keys to its storage file when the file did not exist. To ensure that keys are flushed only when the correct storage file exists, keyring_file now stores a SHA-256 checksum of the keyring in the file. Before updating the file, the plugin verifies that it contains the expected checksum. (Bug #23498254)

START GROUP REPLICATION uses stacked Srv_session and did not return to the correct thread. START GROUP REPLICATION and STOP GROUP REPLICATION are now removed from the list of permitted commands. (Bug #23337984)

A union query resulting in tuples larger than max_join_size could result in a server exit. (Bug #23303485)
• Grant tables with incorrect structure may cause problems in user management operations. As a consequence of the fix for this, for any operation that modifies a grant table, the server now checks whether the table has the expected structure and produces an error if not. `mysql_upgrade` must be run to update the tables to the expected structure. (Bug #23295423, Bug #25095876, Bug #25448037)

• `ST_ExteriorRing()` could cause a server exit due to being passed an invalid WKB string believed to be valid. (Bug #23280574)

• For debug builds, failure to prepare a branch of an XA transaction could lead to a server exit. (Bug #23264552, Bug #81375)

• The optimizer could choose `ref` access on a secondary index rather than `range` access on the primary key, even when the cost was higher. (Bug #23259872, Bug #81341)

• An instance of the disk-full error message contained the wrong error code. (Bug #23247332, Bug #81346)

• For a query with `ORDER BY` and `LIMIT`, an optimizer trace did not record the optimizer’s switch to a different index. (Bug #23227428, Bug #81250)

• Improper handling of a lock used by the `version_tokens` plugin and user-defined functions could result in a server exit if a UDF was called while `version_tokens` was being uninstalled. (Bug #23210850)

• Certain errors in DML statements executed within stored programs could be mishandled and result in a server exit. (Bug #23209989)

• The `QUOTE()` function could allocate excessive memory. A limit of `max_allowed_packet` bytes is now imposed and returns `NULL` with a warning for attempts to allocate more. (Bug #23195404)

• For some deeply nested expressions, the optimizer failed to detect stack overflow, resulting in a server exit. (Bug #23135667)

• For sessions created through the X Plugin, incorrect thread attachment/detachment could cause a server exit. (Bug #23057045)

• When attempting to locate the data directory, `mysqld_safe` incorrectly considered `$MY_BASEDIR_VERSION/var` as one of the possible locations. (Bug #23013510, Bug #80866)

• For some generated columns, character set conversion of the column definition for table rebuilds could change column values. (Bug #22991924)

• The OS X DMG installer did not properly set up keyring plugin installation. (Bug #22991650)

• If a query performed a `GROUP BY` on a column of a derived table and the select list contained an expression mixing an aggregate function and the group column, an error was raised if the `ONLY_FULL_GROUP_BY` SQL mode was enabled. (Bug #22924183, Bug #80726)

• A binary (in-place) upgrade from MySQL 5.6 to 5.7 followed by a data export performed using `mysqlpump` resulted in an `Invalid default value for date_column` error for attempts to reload the dump file. (Bug #22919028, Bug #80706)

• A failed cast of a long float to integer could cause a server exit. (Bug #22907691)

• On Unix and Unix-like systems, the error log file resulting from specifying `--log-error` without an option value was incorrectly created in the directory of the PID file if the `--pid-file` option was also given. (Bug #22900354)
• The `main.log_tables-big` test case could be unstable on highly loaded hosts. Thanks to Laurynas Biveinis for the patch. (Bug #22874167, Bug #80607)

• SQL statements executed through the X Plugin were not instrumented in the Performance Schema. (Bug #22859462)

• The `rpl.rpl_key_rotation` test case did not synchronize properly with the master server. Thanks to Laurynas Biveinis for the patch. (Bug #22838596, Bug #80531)

• `DROP INDEX` operations could fail due to inconsistent handling of index prefix lengths for `TEXT`-type columns (`TINYTEXT` and so forth).

  A consequence of this fix is more restrictive behavior for `CREATE TABLE` and `CREATE INDEX` statements for which a specified index prefix exceeds the maximum column data type size:

  • For a nonunique index, either an error occurs (if strict SQL mode is enabled), or the index length is reduced to lie within the maximum column data type size and a warning is produced (if strict mode is not enabled).

  • For a unique index, an error occurs regardless of SQL mode because reducing the index length might enable insertion of nonunique entries that do not meet the specified uniqueness requirement.

  (Bug #22740093, Bug #80392)

• Metadata locking on stored routine names was performed on a case-sensitive basis, but routine names are not case sensitive. (Bug #22700385)

• Queries that used an aggregate function with `DISTINCT` could produce incorrect results. (Bug #22686994, Bug #80310)

• The `innodb_numa_interleave` system variable was erroneously available on some systems that were not NUMA-enabled. Thanks to Tomislav Plavcic for the patch.

  `CMake` now sets the default `WITH_NUMA` value based on whether the current platform has `NUMA` support. For platforms without NUMA support, `CMake` behaves as follows:

  • With no NUMA option (the normal case), `CMake` continues normally, producing only this warning: NUMA library missing or required version not available

  • With `-DWITH_NUMA=ON`, `CMake` aborts with this error: NUMA library missing or required version not available

  (Bug #22678436, Bug #80288)

• Certain SQL queries involving complex `WHERE` conditions could cause warnings, memory corruption, or a server exit. (Bug #22671573)

• When taking the server offline, a race condition within the Performance Schema could lead to a server exit. (Bug #22551677)

• On macOS, if a table with an associated trigger was renamed to a new name containing both lowercase and uppercase characters, `DROP TRIGGER` for the trigger resulted in an `ER_NO_SUCH_TABLE` error for the table. (Bug #22512899, Bug #79873)

• In the `MYSQL_FIELD` C API structure, the `org_table` value for derived tables was `*`, which could cause failure for queries that depend on this value. The `org_table` value for views and derived tables now is set as follows: If the column is selected from a view, `org_table` names the view. If the column is selected from a derived table, `org_table` names the base table. If a derived table wraps a view,
org_table still names the base table. If the column is an expression, org_table is the empty string. (Bug #22364401, Bug #79641)

- For RPM packages, the default error-log location in the deployed /etc/my.cnf file differed from the location in the installed logrotate script, causing logrotate to fail. (Bug #22322685)

- The Performance Schema events_statements_summary_by_digest table could contain multiple rows for the same statement digest and schema combination, rather than the expected single (unique) row. (Bug #22320066, Bug #79533)

- For Performance Schema system and status variable tables, variable values expressed in a character set different from utf8 could be truncated or incorrect. (Bug #22313205)

- For debug builds, altering a table partitioning expression using an expression attribute with an invalid UTF-8 name caused an assertion to be raised. (Bug #22152229)

- After performing inserts in a table containing an AUTO_INCREMENT column and then performing a SELECT operation, the LAST_INSERT_ID() returns the correct value, but the value of the mysql_insert_id() C API function was being reset to 0. (Bug #22028117, Bug #78778)

- On Ubuntu, error messages were displayed during upgrades from Community to Commercial packages that made it appear as though mysqld and my_print_defaults had not been installed. Those messages were spurious and have been silenced. (Bug #21807248)

- With the use_index_extensions flag of the optimizer_switch system variable disabled, some SELECT DISTINCT queries could return incorrect results. (Bug #21749123, Bug #78244)

- An invalid string value in the WHERE clause of an UPDATE statement, caused an index scan rather than a range scan to be used. For values not present in the index, this could be much slower. Now the optimizer determines this to be an “impossible WHERE” condition. (Bug #21032418, Bug #76933)

- Debian packages were missing an AppArmor-related include file and incorrectly were marked dependent on AppArmor (making it impossible to disable AppArmor by uninstalling it). (Bug #20768958)

- When the automatic_sp_privileges system variable was enabled, it did not have the expected effect for anonymous users. (Bug #20266641)

- In a replication environment, SET PASSWORD or ALTER USER could fail to execute on the slave due to failure to parse the hash string correctly. (Bug #20228478)

- The optimizer resolve_const_item() function called Item_decimal() with the last two arguments in the wrong order. (Bug #19062566)

- An in-place ALTER TABLE operation failed to report an error when adding a DATE or DATETIME column under these conditions: a) the column was NOT NULL and no default value was supplied; b) strict and NO_ZERO_DATE SQL modes were enabled; c) the table was not empty.

An ALTER TABLE operation failed with an error rather than a warning when adding a DATE or DATETIME column under these conditions: a) the column was NOT NULL and no default value was supplied; b) strict SQL mode was enabled and NO_ZERO_DATE SQL mode was not enabled; c) the table was not empty. (Bug #16888677)

- Inserting a TIME, DATE, or TIMESTAMP value with a fractional seconds part into a column having the same type but fewer fractional digits resulted in rounding. This differs from MySQL 5.5, which used truncation rather than rounding. To enable control over this behavior, a new TIME_TRUNCATE_FRACTIONAL SQL mode is available. The default is to use rounding. If this mode is enabled, truncation occurs instead. (Bug #16583910, Bug #68760)
• **NOT IN** subqueries could produce incorrect results when an index prefix of an inner table was used. (Bug #13915291)

• On non-Linux Unix systems, the `mysql.server` startup script used the Linux command `pidof` rather than `pgrep`. (Bug #13788154, Bug #64342)

• Starting multiple instances of `mysqld_safe` after an abnormal server exit could result in one `mysqld_safe` instance killing another. As a consequence of the bug fix, the `mysqld_safe.pid` file is no longer used. (Bug #11751149, Bug #41908)

• The `--help` message for `mysqld_safe` was corrected to mention that the `--no-defaults`, `--defaults-file`, and `--defaults-extra-file` options, if given, must be the first argument. (Bug #11745176, Bug #11192)

• The bounds check for the XML parser position stack for each level (which has a fixed depth) used the size of the array as the upper limit, and so was off by one. This is fixed by decreasing the allowable depth by one, which actually matches the maximum number of elements in the position stack. (Bug #83871, Bug #25111907)

References: See also: Bug #14040071, Bug #15948580.

### Changes in MySQL 8.0.0 (2016-09-12, Development Milestone)

**Note**

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)

• Account Management Notes
• C API Notes
• Character Set Support
• Compilation Notes
• Component Notes
• Configuration Notes
• Data Dictionary Notes
• Data Type Notes
• Doxygen Notes
• Optimizer Notes
• Packaging Notes
• Parser Notes
• **Performance Schema Notes**
• **Security Notes**
• **Spatial Data Support**
• **Test Suite Notes**
• **X Plugin Notes**
• **Functionality Added or Changed**
• **Bugs Fixed**

**Account Management Notes**

**Incompatible Change:** The grant tables in the mysql system database are now InnoDB (transactional) tables. Previously, these were MyISAM (nontransactional) tables. This change applies to these tables: `user, db, tables_priv, columns_priv, procs_priv, proxies_priv`.

The change of grant table storage engine underlies an accompanying change to the behavior of account-management statements. Previously, an account-management statement that named multiple users could succeed for some users and fail for others. Now, each statement is transactional and either succeeds for all named users or rolls back and has no effect if any error occurs. The statement is written to the binary log if it succeeds, but not if it fails; in that case, rollback occurs and no changes are made. The preceding behavior applies to these statements: `ALTER_USER`, `CREATE_ROLE`, `CREATE_USER`, `DROP_ROLE`, `DROP_USER`, `GRANT`, `RENAME_USER`, `REVOKE`. (SET_PASSWORD is not listed because it applies to at most one user and is effectively transactional already.) A side effect of this change in behavior is that partially completed account management statements on a MySQL 5.7 master fail when replicated on a MySQL 8.0 slave. For more information, see Atomic Data Definition Statement Support.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the mysql system database.

---

**Note**

If MySQL is upgraded from an older version but the grant tables have not been upgraded from MyISAM to InnoDB, the server considers them read only and account-management statements produce an error.

Due to the change of storage engine from MyISAM to InnoDB, `SELECT` without `ORDER BY` on grant tables can produce different row orders than previously. If a query result must have specific row ordering characteristics, include an `ORDER BY` clause.

• MySQL now supports roles, which are named collections of privileges. Roles enable assignment of sets of privileges to accounts and provide a convenient alternative to granting individual privileges, both for conceptualizing desired privilege assignments and implementing them:
  • Roles can be created and dropped.
  • Roles can have privileges granted to and revoked from them.
  • Roles can be granted to and revoked from user accounts.
  • The active roles for an account can be selected from among those granted to the account, and can be changed during sessions for that account.

For more information, see Using Roles.
**Note**

ROLE now is a reserved word and cannot be used as an identifier without identifier quoting.

**C API Notes**

- The *libmysqlclient* shared library major version number is increased from 20 (used in MySQL 5.7) to 21 for MySQL 8.0. (Bug #77600, Bug #21363863)

**Character Set Support**

- The *utf8mb4* Unicode character set has a new general collation named *utf8mb4_0900_ai_ci*. *utf8mb4* also has several new language-specific collations with characteristics similar to *utf8mb4_0900_ai_ci* except that language-specific rules take precedence where applicable. The language-specific collations are indicated by ISO 639-1 language codes in the collation name, as shown in the following table. In two cases the language code has an additional item that denotes a variant (German phone book order, Traditional Spanish).

<table>
<thead>
<tr>
<th>Language</th>
<th>Collation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Croatian</td>
<td><em>utf8mb4_hr_0900_ai_ci</em></td>
</tr>
<tr>
<td>Czech</td>
<td><em>utf8mb4_cs_0900_ai_ci</em></td>
</tr>
<tr>
<td>Danish</td>
<td><em>utf8mb4_da_0900_ai_ci</em></td>
</tr>
<tr>
<td>Esperanto</td>
<td><em>utf8mb4_eo_0900_ai_ci</em></td>
</tr>
<tr>
<td>Estonian</td>
<td><em>utf8mb4_et_0900_ai_ci</em></td>
</tr>
<tr>
<td>German phone book order</td>
<td><em>utf8mb4_de_pb_0900_ai_ci</em></td>
</tr>
<tr>
<td>Hungarian</td>
<td><em>utf8mb4_hu_0900_ai_ci</em></td>
</tr>
<tr>
<td>Icelandic</td>
<td><em>utf8mb4_is_0900_ai_ci</em></td>
</tr>
<tr>
<td>Latvian</td>
<td><em>utf8mb4_lv_0900_ai_ci</em></td>
</tr>
<tr>
<td>Lithuanian</td>
<td><em>utf8mb4_lt_0900_ai_ci</em></td>
</tr>
<tr>
<td>Polish</td>
<td><em>utf8mb4_pl_0900_ai_ci</em></td>
</tr>
<tr>
<td>Classical Latin</td>
<td><em>utf8mb4_la_0900_ai_ci</em></td>
</tr>
<tr>
<td>Romanian</td>
<td><em>utf8mb4_ro_0900_ai_ci</em></td>
</tr>
<tr>
<td>Slovak</td>
<td><em>utf8mb4_sk_0900_ai_ci</em></td>
</tr>
<tr>
<td>Slovenian</td>
<td><em>utf8mb4_sl_0900_ai_ci</em></td>
</tr>
<tr>
<td>Modern Spanish</td>
<td><em>utf8mb4_es_0900_ai_ci</em></td>
</tr>
<tr>
<td>Traditional Spanish</td>
<td><em>utf8mb4_es_trad_0900_ai_ci</em></td>
</tr>
<tr>
<td>Swedish</td>
<td><em>utf8mb4_sv_0900_ai_ci</em></td>
</tr>
<tr>
<td>Turkish</td>
<td><em>utf8mb4_tr_0900_ai_ci</em></td>
</tr>
<tr>
<td>Vietnamese</td>
<td><em>utf8mb4_vi_0900_ai_ci</em></td>
</tr>
</tbody>
</table>

*utf8mb4_0900_ai_ci* also works as an accent- insensitive, case-insensitive collation for the languages in the following table.
Table 6 Languages for which utf8mb4_0900_ai_ci is suitable

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>German (dictionary order)</td>
<td>de</td>
</tr>
<tr>
<td>English</td>
<td>en</td>
</tr>
<tr>
<td>Canadian French (locale fr_CA)</td>
<td>fr</td>
</tr>
<tr>
<td>Irish Gaelic</td>
<td>ga</td>
</tr>
<tr>
<td>Indonesian</td>
<td>id</td>
</tr>
<tr>
<td>Italian</td>
<td>it</td>
</tr>
<tr>
<td>Luxembourgian</td>
<td>lb</td>
</tr>
<tr>
<td>Malay</td>
<td>ms</td>
</tr>
<tr>
<td>Dutch</td>
<td>nl</td>
</tr>
<tr>
<td>Portuguese</td>
<td>pt</td>
</tr>
<tr>
<td>Swahili</td>
<td>sw</td>
</tr>
<tr>
<td>Zulu</td>
<td>zu</td>
</tr>
</tbody>
</table>

utf8mb4_da_0900_ai_ci also works as an accent-insensitive, case-insensitive collation for the languages in the following table.

Table 7 Languages for Which utf8mb4_da_0900_ai_ci is Suitable

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Norwegian</td>
<td>no</td>
</tr>
<tr>
<td>Norwegian Bokmål</td>
<td>nb</td>
</tr>
<tr>
<td>Norwegian Nynorsk</td>
<td>nn</td>
</tr>
</tbody>
</table>

The nonlanguage-specific utf8mb4_0900_ai_ci and language-specific utf8mb4_LANG_0900_ai_ci Unicode collations each have these characteristics:

- The collation is based on Unicode Collation Algorithm (UCA) 9.0.0 and Common Locale Data Repository (CLDR) v30, is accent insensitive, and case insensitive. These characteristics are indicated by _0900, _ai, and _ci in the collation name. Exception: utf8mb4_la_0900_ai_ci is not based on CLDR because Classical Latin is not defined in CLDR.

- The collation works for all characters in the range [U+0, U+10FFFF].

- If the collation is not language specific, it sorts all characters, including supplemental characters, in default order (described following). If the collation is language specific, it sorts characters of the language correctly according to language-specific rules, and characters not in the language in default order.

- By default, the collation sorts characters having a code point listed in the DUCET table (Default Unicode Collation Element Table) according to the weight value assigned in the table. The collation sorts characters not having a code point listed in the DUCET table using their implicit weight value, which is constructed according to the UCA.

- For non-language-specific collations, characters in contraction sequences are treated as separate characters. For language-specific collations, contractions might change character sorting order.
For more information, see Unicode Character Sets.

Compilation Notes

- **Microsoft Windows:** For building MySQL on Windows, the toolchain now prefers 64-bit tools when possible (previously 32-bit). This speeds up linking and avoids issues related to limited address space with the 32-bit linker. (Bug #80675, Bug #22900585)

- **CMake** now causes the build process to link with the GNU gold linker if it is available and not explicitly disabled. To disable use of this linker, specify the `-DUSE_LD_GOLD=OFF` CMake option. (Bug #23759968, Bug #82163)

- The **WITH_EXTRA_CHARSET** CMake option has been removed. MySQL builds are configured with all character sets by default now. Users who want fewer character sets can edit `cmake/character_sets.cmake` directly and recompile the server. (Bug #80005, Bug #22552125)

- The minimum version of the Boost library for server builds is now 1.60.0. (Bug #79380, Bug #22253921)

- Work was done to clean up the source code base, including: Removing unneeded CMake checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

- MySQL source code now permits and uses C++11 features. To enable a good level of C++11 support across all supported platforms, the following minimum compiler versions now apply:
  - GCC: 4.8 or higher
  - Clang: 3.4 or higher (Xcode 7 on OS X)
  - Solaris Studio: 12.4 or higher (Solaris client build only)
  - Visual Studio: 2015
  - CMake: On Windows, the required Visual Studio version results in a required CMake version of 3.2.3 or higher

  On Solaris, the stlport library is no longer used. This makes the `SUNPRO_CXX_LIBRARY` CMake option obsolete, so it has been removed.

Component Notes

- MySQL Server now includes a component-based infrastructure for improving server extensibility:
  - A component provides services that are available to the server and other components. (With respect to service use, the server is a component, equal to other components.) Components interact with each other only through the services they provide.
  - The `INSTALL COMPONENT` and `UNINSTALL COMPONENT` statements provide an SQL interface for component manipulation at runtime.
  - A loader service registers installed components in the `mysql.component` system table, and installs registered components during the startup sequence for subsequent server restarts.

  For general information about the component infrastructure and its SQL-level interface, see MySQL Server Components. For information about the internal implementation of components, see the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html.
Configuration Notes

- **Incompatible Change; InnoDB:** Previously, enabling the `innodb_read_only` system variable prevented creating and dropping tables only for the InnoDB storage. As of MySQL 8.0, enabling `innodb_read_only` prevents these operations for all storage engines. Table creation and drop operations modify data dictionary tables in the `mysql` system database, but those tables use the InnoDB storage engine and cannot be modified when `innodb_read_only` is enabled. The same principle applies to other table operations that require modifying data dictionary tables, and to operations that modify other tables in the `mysql` database that use the InnoDB storage engine, such as the grant tables and the `func` and `plugin` tables. (Bug #21611899)

- The hardcoded memory page size of 8KB for the memory-mapped transaction coordinator was too small for platforms such as ARM64 and PowerPC where the page size is much larger. The server now invokes a system call to get the page size of the current platform rather than using a hardcoded value. A consequence for the `--log-tc-size` option is that the minimum and default values are now 6 times the page size. Also, the value must be a multiple of the page size. Thanks to Alexey Kopytov for the patch. (Bug #23014086, Bug #80818, Bug #26931470, Bug #87995)

- MySQL now supports a `SET PERSIST` variant of `SET` statement syntax, for making configuration changes at runtime that also persist across server restarts. Like `SET GLOBAL`, `SET PERSIST` is permitted for any global system variable that is dynamic (settable at runtime). The statement changes the runtime variable value, but also writes the variable setting to an option file named `mysqld-auto.cnf` in the data directory. At startup, the server processes this file after all other option files. For more information, see Persisted System Variables.

To provide information showing how each system variable was most recently set, the Performance Schema now has a `variables_info` table that lists each system variable and the source from which it got its value. See Performance Schema variables_info Table.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this change into the Performance Schema.

- The deprecated `mysql_install_db` program has been removed from MySQL distributions. Data directory initialization should be performed by invoking `mysqld` with the `--initialize` or `--initialize-insecure` option instead. In addition, the deprecated `--bootstrap` option for `mysqld` that was used by `mysql_install_db` has been removed, and the `INSTALL_SCRIPTDIR` CMake option that controlled the installation location for `mysql_install_db` has been removed.

Version 1 test suite code previously was located in the `mysql-test/lib/v1` directory of MySQL source distributions. This code used `mysql_install_db` and has been removed. The `MYSQL_INSTALL_DB` environment variable and a value of 1 for the `MTR_VERSION` environment variable are no longer supported.

Data Dictionary Notes

- **Incompatible Change:** MySQL Server now incorporates a global data dictionary containing information about database objects in transactional tables. In previous MySQL releases, dictionary data was stored in metadata files and nontransactional system tables.

  **Important**

  A data dictionary-enabled server entails some general operational differences compared to a server that does not have a data dictionary; see Data Dictionary Usage Differences. Also, for upgrades to MySQL 8.0, the upgrade procedure differs somewhat from previous MySQL releases and requires that you verify the upgrade readiness of your installation by checking specific prerequisites. For
more information, see Upgrading MySQL, particularly Preparing Your Installation for Upgrade.

**InnoDB** continues to use its own data dictionary in the MySQL 8.0.0 release.

The following list briefly describes the main implications of this change:

- The `.frm` metadata files previously associated with base tables and views no longer exist. Metadata previously stored in `.frm` files is now stored in data dictionary tables.

  Similarly, trigger metadata previously stored in `.TRG` and `.TRN` files is stored in a data dictionary table and those files no longer exist.

- With the removal of `.frm` files, the 64KB table definition size limit imposed by the `.frm` file structure is removed.

- With the removal of `.frm` files, the `INFORMATION_SCHEMA.TABLES` `VERSION` field now reports a hardcoded value of 10, which is the last `.frm` file version used in MySQL 5.7.

- A new dictionary object cache that serves the MySQL data dictionary stores previously accessed data dictionary objects in memory to enable object reuse and minimize disk I/O. An LRU-based eviction strategy is used to evict least recently used objects from memory. The cache comprises several partitions that store different object types. For more information, see Dictionary Object Cache.

- New internal data dictionary APIs enable the server, internal storage engines, and plugins to access and store data in the MySQL data dictionary. Internal data dictionary APIs are introduced for handling of schemas, tablespaces, tablespace files, tables, partitioned tables, table partition data, triggers, stored routines, events, table objects, views, character sets, and collations.

  With this change, data dictionary updates and binary log writes for `CREATE TRIGGER` and `DROP TRIGGER` operations are combined into a single, atomic transaction.

- Data dictionary tables are invisible, but in most cases there are corresponding `INFORMATION_SCHEMA` tables that can be queried instead. This enables the underlying data dictionary tables to be changed as server development proceeds, while maintaining a stable `INFORMATION_SCHEMA` interface for application use.

  Some `INFORMATION_SCHEMA` tables have been reimplemented entirely as views on data dictionary tables:

<table>
<thead>
<tr>
<th>CHARACTER_SETS</th>
<th>COLLATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLLATION_CHARACTER_SET_APPLICABILITY</td>
<td>COLUMNS</td>
</tr>
</tbody>
</table>

  Queries on those tables are now more efficient because they obtain information from data dictionary tables rather than by other, slower means. In particular, for each `INFORMATION_SCHEMA` table that is a view on data dictionary tables:

  - The server no longer must create a temporary table for each query of the `INFORMATION_SCHEMA` table.
• When the underlying data dictionary tables store values previously obtained by directory
scans (for example, to enumerate database names or table names within databases) or file-
opening operations (for example, to read information from .frm files), INFORMATION_SCHEMA
queries for those values now use table lookups instead. (Additionally, even for a non-view
INFORMATION_SCHEMA table, values such as database and table names are retrieved by lookups
from the data dictionary and do not require directory or file scans.)

• Indexes on the underlying data dictionary tables permit the optimizer to construct efficient
query execution plans, something not true for the previous implementation that processed the
INFORMATION_SCHEMA table using a temporary table per query.

The preceding improvements also apply to SHOW statements that display information corresponding
to the INFORMATION_SCHEMA tables that are views on data dictionary tables. For example, SHOW DATABASES displays the same information as the SCHEMATA table.

For INFORMATION_SCHEMA queries that retrieve table statistics, the server now can use statistics
cached in INFORMATION_SCHEMA tables, or obtain the latest statistics directly from storage engines.
The information_schema_stats system variable controls which statistics source the server uses.

• When information_schema_stats is CACHED (the default), the server uses cached statistics
stored in the STATISTICS and TABLES tables.

• When information_schema_stats is LATEST, the server obtains statistics directly from storage
engines. In this case, the server treats queries on STATISTICS and TABLES as queries for the
latest statistics stored in the STATISTICS_DYNAMIC and TABLES_DYNAMIC tables.

Affected INFORMATION_SCHEMA table statistic columns include:

- STATISTICS.CARDINALITY
- TABLES.AUTO_INCREMENT
- TABLES.AVG_ROW_LENGTH
- TABLES.CHECKSUM
- TABLES.CHECK_TIME
- TABLES.CREATE_TIME
- TABLES.DATA_FREE
- TABLES.DATA_LENGTH
- TABLES.INDEX_LENGTH
- TABLES.MAX_DATA_LENGTH
- TABLES.TABLE_ROWS
- TABLES.UPDATE_TIME

For more information, see Optimizing INFORMATION_SCHEMA Queries.

• The foreign_keys and foreign_key_column_usage tables now store foreign key information.
The standard SQL way to obtain foreign key information is by using the INFORMATION_SCHEMA
REFERENTIAL_CONSTRAINTS and KEY_COLUMN_USAGE tables; these tables are now implemented
as views on the foreign_keys, foreign_key_column_usage, and other data dictionary tables.

For some foreign key errors, the server now produces more appropriate and more informative error
messages.

![Note]

Incompatibility: Previously, MySQL supported foreign key names longer
than 64 characters. Foreign key names as stored in the foreign_keys and
foreign_key_column_usage tables are a maximum of 64 characters, per the SQL standard, so longer foreign key names are no longer permitted.

- Because the data dictionary provides information about database objects, the server no longer checks directory names in the data directory to find databases. Consequently, the --ignore-db-dir option and ignore_db_dirs system variable are extraneous and have been removed. Update system configurations and application programs accordingly.

- System table changes:

  - Many system tables have been converted from MyISAM (nontransactional) tables to InnoDB (transactional) tables. For example, as discussed elsewhere in these release notes, the grant tables are now InnoDB tables. Other examples follow.

  - The func table that stores user-defined function information in the mysql system database now is an InnoDB (transactional) table. Previously, it was a MyISAM (nontransactional) table.

    In consequence of this change, CREATE FUNCTION and DROP FUNCTION statements cause an implicit commit, even when used for user-defined functions (see Statements That Cause an Implicit Commit). Previously, they caused an implicit commit when used for stored functions, but not for user-defined functions.

  - Previously, information about stored routines and events was stored in the proc and event tables of the mysql system database. Those tables are no longer used. Instead, information about stored routines and events is stored in the routines, events, and parameters data dictionary tables in the mysql system database. The old tables used the MyISAM (nontransactional) storage engine. The new tables use the InnoDB (transactional) engine.

    Previously, creating a stored routine that contained illegal characters produced a warning. This is now an error.

  - To permit access to system tables (for example, time zone or log tables) to be distinguished from access to nonsystem tables, the server uses the Locking system tables and Opening system tables thread states rather than the System lock and Opening tables thread states. See General Thread States.

- InnoDB changes:

  - Persistent InnoDB tablespaces now include transactional storage for Serialized Dictionary Information (SDI), which is dictionary object data in serialized form. Along with the disappearance of .frm and trigger metadata files, mentioned previously, you might notice the appearance of .SDI files. These are serialized dictionary information files. SDI transactional storage is reserved for an in-progress feature not yet fully implemented.

  - A new command-line utility, ibd2sdi, is used to extract serialized dictionary information (SDI) from persistent InnoDB tablespaces. SDI data is not present in persistent InnoDB tablespaces in this release. The ibd2sdi utility is reserved for future use.

  - InnoDB startup code was refactored to support MySQL initialization changes related to the MySQL data dictionary feature.

- Upgrade and downgrade implications:

  - To upgrade to MySQL 8.0 from MySQL 5.7, you must perform the upgrade procedure described at Upgrading MySQL.
• Downgrading from MySQL 8.0 to MySQL 5.7 is only supported using the logical downgrade method (a mysqldump downgrade). In-place downgrades are not supported.

(Bug #80481, Bug #22811659)

Data Type Notes

• Bit functions and operators comprise `BIT_COUNT()`, `BIT_AND()`, `BIT_OR()`, `BIT_XOR()`, `&`, `|`, `^`, `~`, `<<`, and `>>`. Prior to MySQL 8.0, bit functions and operators required `BIGINT` (64-bit integer) arguments and returned `BIGINT` values, so they had a maximum range of 64 bits. Non-`BIGINT` arguments were converted to `BIGINT` prior to performing the operation and truncation could occur. Now bit functions and operators permit binary string type arguments (`BINARY`, `VARBINARY`, and the `BLOB` types) and return a value of like type, which enables them to take arguments and produce return values larger than 64 bits. Nonbinary string arguments are converted to `BIGINT` and processed as such, as before.

Permitting binary string arguments for bit functions and operators makes it easier not only to manipulate larger values, but to perform bit operations not easily done previously on certain types of data, such as UUID and IPv6 values. For examples, see Bit Functions and Operators.

An implication of this change in behavior is that bit operations on binary string arguments might produce a different result in MySQL 8.0 than in 5.7. For information about how to prepare in MySQL 5.7 for potential incompatibilities between MySQL 5.7 and 8.0, see Bit Functions and Operators, in MySQL 5.7 Reference Manual.

Doxygen Notes

• The MySQL source code has been updated to use Doxygen for the internal documentation. This is a work in progress. As new MySQL versions are distributed, the Doxygen documentation will be updated, with the latest version always available at https://dev.mysql.com/doc/index-other.html.

It is also possible to generate the Doxygen content locally from a MySQL source distribution using the instructions at Generating MySQL Doxygen Documentation Content.

Optimizer Notes

• **InnoDB:** The storage engine interface now enables the optimizer to provide information about the size of the record buffer to be used for scans that the optimizer estimates will read multiple rows. The buffer size can vary based on the size of the estimate. InnoDB uses this variable-size buffering capability to take advantage of row prefetching, and to reduce the overhead of latching and B-tree navigation. Previously, InnoDB used a small, fixed-size buffer.

• The optimizer now supports table-level `MERGE` and `NO_MERGE` hints for specifying whether derived tables or views should be merged into the outer query block or materialized using an internal temporary table. Examples:

```sql
SELECT /*+ MERGE(dt) */ * FROM (SELECT * FROM t1) AS dt;
SELECT /*+ NO_MERGE(dt) */ * FROM (SELECT * FROM t1) AS dt;
```

For more information, see Optimizer Hints. (Bug #79554, Bug #22328100)

• MySQL now supports invisible indexes. An invisible index is not used by the optimizer at all, but is otherwise maintained normally. Indexes are visible by default. Invisible indexes make it possible to test the effect of removing an index on query performance, without making a destructive change that must be undone should the index turn out to be required. This feature applies to InnoDB tables, for indexes other than primary keys.
To control whether an index is invisible explicitly for a new index, use a `VISIBLE` or `INVISIBLE` keyword as part of the index definition for `CREATE TABLE`, `CREATE INDEX`, or `ALTER TABLE`. To alter the invisibility of an existing index, use a `VISIBLE` or `INVISIBLE` keyword with the `ALTER TABLE` ... `ALTER INDEX` operation. For more information, see Invisible Indexes.

- The `mysql` system database now contains a `column_stats` table designed to store statistics about column values. For more information, see Optimizer Statistics.

Packaging Notes

- Development milestone releases in previous MySQL series were numbered using a suffix of `-mN`, to indicate development milestone `N`. In MySQL 8.0, development releases use the suffix `-dmr`. For example, this release of MySQL is numbered `8.0.0-dmr`. (Bug #80408, Bug #22748154)
- As a consequence of the use of C++11 features described elsewhere in these release notes, the following packaging changes have been made:
  - Support for Red Hat Enterprise Linux 5 and Oracle Linux 5 RPMs has been dropped
  - Generic binary tarball builds have been moved to Red Hat Enterprise Linux 6

Parser Notes

- The parser rules for `SELECT` and `UNION` were refactored to be more consistent (the same `SELECT` syntax applies uniformly in each such context) and reduce duplication. Several user-visible effects resulted from this work:
  - `NATURAL JOIN` permits an optional `INNER` keyword (`NATURAL INNER JOIN`), in compliance with standard SQL.
  - Right-deep joins without parentheses are permitted (for example, `... JOIN ... JOIN ... ON ... ON`), in compliance with standard SQL.
  - `STRAIGHT_JOIN` now permits a `USING` clause, similar to other inner joins.
  - The parser accepts parentheses around query expressions. For example, `(SELECT ... UNION SELECT ...)` is permitted.
  - The parser better conforms to the documented permitted placement of the `SQL_CACHE` and `SQL_NO_CACHE` query modifiers.
  - Left-hand nesting of unions, previously permitted only in subqueries, is now permitted in top-level statements. For example, this statement is now accepted as valid:

    ```sql
    (SELECT 1 UNION SELECT 1) UNION SELECT 1;
    ```

    (Bug #11746363, Bug #25734)

- The parser rules for `CREATE TABLE` were refactored to be context independent and improve maintainability and extensibility. Several user-visible effects resulted from this work:
  - For generated columns, including `NOT NULL NULL` resulted in a column that included the `NOT NULL` attribute, which differed from nongenerated columns. Such definitions now use the final attribute `NULL`, resulting in a nullable column (consistent with nongenerated columns).
  - `CREATE TEMPORARY TABLE` no longer permits multiple instances of `TEMPORARY`. 


• Previously, `PARSE_GCOL_EXPR` was a keyword and could not be used as a label in stored programs. It is no longer a keyword and can be used as a label.

• Messages for some syntax errors are more precise with respect to the location of the error within the statement.

### Performance Schema Notes

**Incompatible Change:** The Performance Schema now instruments server errors (and warnings), and exposes statistical information about them through a set of summary tables:

• The **error** instrument controls whether error information is collected (enabled by default).

• Several tables contain error information, summarized in various ways:
  
  - `events_errors_summary_global_by_error`
  - `events_errors_summary_by_account_by_error`
  - `events_errors_summary_by_host_by_error`
  - `events_errors_summary_by_thread_by_error`
  - `events_errors_summary_by_user_by_error`.

• The **performance_schema_error_size** system variable controls the number of instrumented errors.

For more information, see [Error Summary Tables](#)

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `performance_schema` database.

In consequence of the preceding changes, two server error symbols were renamed:

- `ER_CANT_SET_ENFORCE_GTID_CONSISTENCY_ON_WITHONGOING_GTID_VIOLATING_TRANSACTIONS` is now `ER_CANT_ENFORCE_GTID_CONSISTENCY_WITHONGOING_GTID_VIOLATING_TX` and
- `ER_SET_ENFORCE_GTID_CONSISTENCY_WARN_WITHONGOING_GTID_VIOLATING_TRANSACTIONS` is now `ER_ENFORCE_GTID_CONSISTENCY_WARN_WITHONGOING_GTID_VIOLATING_TX`.

Also, several server error codes were found to be no longer used in the server and have been removed as obsolete. Applications that test specifically for any of these errors should be updated:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ER_JUMP_OVERRIDDEN</td>
<td>This error is raised when a stored procedure is executed and an exception is generated at a point where the <code>EXCEPTION</code> keyword is used to override a normal return value.</td>
</tr>
<tr>
<td>ER_TRUNCATED_DATA</td>
<td>This error is raised when data is too large to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_COLUMNS</td>
<td>This error is raised when a column is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL</td>
<td>This error is raised when a SQL statement is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_COLUMN</td>
<td>This error is raised when a column is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_TABLE</td>
<td>This error is raised when a table is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_INDEX</td>
<td>This error is raised when an index is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_COLUMN_INDEX</td>
<td>This error is raised when a column index is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_TABLE_INDEX</td>
<td>This error is raised when a table index is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_COLUMN_TABLE_INDEX</td>
<td>This error is raised when a column table index is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_TABLE_COLUMN_INDEX</td>
<td>This error is raised when a table column index is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_TABLE_COLUMN_TABLE_INDEX</td>
<td>This error is raised when a table column table index is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_INDEX_TABLE_COLUMN_INDEX</td>
<td>This error is raised when an index table column index is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
<tr>
<td>ER_TRUNCATED_SQL_INDEX_TABLE_COLUMN_TABLE_INDEX</td>
<td>This error is raised when an index table column table index is too long to be stored in an internal buffer and can't be stored in a column.</td>
</tr>
</tbody>
</table>

For more information, see [Error Summary Tables](#)
ER_ERROR_DURING_CHECKPOINT
ER_ERROR_ON_CLOSE
ER_EVENTS_DB_ERROR
ER_EVENT_CANNOT_DELETE
ER_EVENT_CANT_ALTER
ER_EVENT_COMPILE_ERROR
ER_EVENT_DATA_TOO_LONG
ER_EVENT_DROP_FAILED
ER_EVENT_MODIFY_QUEUE_ERROR
ER_EVENT_NEITHER_M_EXPR_NOR_M_AT
ER_EVENT_OPEN_TABLE_FAILED
ER_EVENT_STORE_FAILED
ER_EXEC_STMT_WITH_OPEN_CURSOR
ER_FAILED_ROUTINE_BREAK_BINLOG
ER_FLUSH_MASTER_BINLOG_CLOSED
ER_FORM_NOT_FOUND
ER_FOUND_GTID_EVENT_WHEN_GTID_MODE_IS_OFF_UNUSED
ER_FRM_UNKNOWN_TYPE
ER_GOT_SIGNAL
ER_GRANT_PLUGIN_USER_EXISTS
ER_GTID_MODE_REQUIRES_BINLOG
ER_GTID_NEXT_IS_NOT_IN_GTID_NEXT_LIST
ER_HASHCHK
ER_INDEX_REBUILD
ER_INNODB_NO_FTUSES_PARSER
ER_LIST_OF_FIELDSONLY_IN_HASH_ERROR
ER_LOAD_DATA_INVALID_COLUMN_UNUSED
ER_LOGGING_PROHIBIT_CHANGING_OF
ER_MALFORMED_DEFINER
ER_MASTER_KEY_ROTATION_ERROR_BY_SE
ER_NDB_CANT_SWITCH_BINLOG_FORMAT
ER_NEVER_USED
ER_NISAMCHK
ER_NO_CONST_EXPR_IN_RANGE_OR_LIST_ERROR
ER_NO_FILE_MAPPING
ER_NO_GROUP_FOR_PROC
ER_NO_RAID_COMPILED
ER_NO_SUCH_KEY_VALUE
ER_NO_SUCH_PARTITIONUNUSED
ER_OBSOLETE_CANNOT_LOAD_FROM_TABLE
ER_OBSOLETE_COL_COUNT_DOESNT_MATCH_CORRUPTED
ER_ORDER_WITH_PROC
ER_PARTITION_SUBPARTITION_ERROR
ER_PARTITION_SUBPART_MIX_ERROR
ER_PART_STATE_ERROR
ER_PASSWD_LENGTH
ER_QUERY_ON_MASTER
ER_RBR_NOT_AVAILABLE
ER_SKIPPING_LOGGED_TRANSACTION
ER_SLAVE_CHANNEL_DELETE
ER_SLAVE_MULTIPLE_CHANNELS_HOST_PORT
ER_SLAVE_MUST_STOP
ER_SLAVE_WAS_NOT_RUNNING
ER_SLAVE_WAS_RUNNING
ER_SP_GOTO_IN_HNDLR
ER_SP_PROC_TABLE_CORRUPT
ER_SQL_MODE_NO_EFFECT
ER_SR_INVALID_CREATION_CTX
ER_TABLE_NEEDS_UPG_PART
ER_TOO_MUCH_AUTO_TIMESTAMP_COLS
ER_UNEXPECTED_EOF
ER_UNION_TABLES_IN_DIFFERENT_DIR
ER_UNSUPPORTED_BY_REPLICATION_THREAD
ER_UNUSED1
ER_UNUSED2
ER_UNUSED3
ER_UNUSED4
• Previously, the `DIGEST` and `DIGEST_TEXT` columns in the Performance Schema `events_statements_current` table were populated only after statement execution ended. Now, the columns are populated just after parsing and before statement execution begins. This enables monitoring applications to access statement digest information during statement execution. (Bug #23336542)

• Previously, Performance Schema optimizations focused on reducing the overhead involved in collecting monitoring data. Complementing that earlier work, overhead now is also reduced for Performance Schema queries that retrieve that data. This is achieved by the addition of indexes to most Performance Schema tables, which gives the optimizer access to execution plans other than full table scans. These indexes also improve performance for related objects, such as `sys` schema views that use those tables. For more information, see Optimizing Performance Schema Queries.

• The size of the `ROLE` column of the `setup_actors` Performance Schema table was increased from 16 to 32 characters.

Security Notes

• The `validate_password_check_user_name` system variable is now enabled by default rather than disabled. This means that when the `validate_password` plugin is enabled, by default it now rejects passwords that match the current session user name.

• The client-side `--ssl` and `--ssl-verify-server-cert` options have been removed. Use `--ssl-mode=REQUIRED` instead of `--ssl=1` or `--enable-ssl`. Use `--ssl-mode=DISABLED` instead of `--ssl=0`, `--skip-ssl`, or `--disable-ssl`. Use `--ssl-mode=VERIFY_IDENTITY` instead of `--ssl-verify-server-cert` options. (The server-side `--ssl` option remains unchanged.)

For the C API, `MYSQL_OPT_SSL_ENFORCE` and `MYSQL_OPT_SSL_VERIFY_SERVER_CERT` options for `mysql_options()` correspond to the client-side `--ssl` and `--ssl-verify-server-cert` options and have been removed. Use `MYSQL_OPT_SSL_MODE` with an option value of `SSL_MODE_REQUIRED` or `SSL_MODE_VERIFY_IDENTITY` instead.

Spatial Data Support

• Spatial functions for import and export of Well-Known Text (WKT) values used MySQL 'GEOMETRYCOLLECTION()' nonstandard syntax rather than OpenGIS 'GEOMETRYCOLLECTION EMPTY' standard syntax. Now both syntaxes are understood for import and the standard syntax is used for export. See Functions That Create Geometry Values from WKT Values. (Bug #23632147, Bug #81964)

• The `ST_X()` and `ST_Y()` spatial functions now permit an optional second argument that specifies an X or Y coordinate value, respectively. With two arguments, the function result is the point value from the first argument with the appropriate coordinate modified. In addition, `ST_X()` and `ST_Y()` with a single argument now are stricter and produce an `ER_UNEXPECTED_GEOMETRY_TYPE` error rather than returning `NULL` if the argument is a valid geometry but not a point. For more information, see Point Property Functions.

• The `ST_SRID()` spatial function now permits an optional second argument that specifies a SRID value. With two arguments, the function result is the geometry value from the first argument with its SRID modified according to the second argument. For more information, see General Geometry Property Functions.
MySQL now stores information about spatial reference systems other than SRID 0, for use with spatial data. This information is stored in the st_spatial_reference_systems data dictionary table and is based on EPSG Dataset 8.7. For information about spatial reference systems, see Spatial Reference System Support.

Previously, the ST_IsValid(), ST_MakeEnvelope(), and ST_Validate() functions required geometry arguments with SRID 0. They now accept geometry arguments with an SRID for a projected spatial reference system.

In MySQL 5.7, several spatial functions available under multiple names were deprecated to move in the direction of making the spatial function namespace more consistent, the goal being that each spatial function name begin with ST_ if it performs an exact operation, or with MBR if it performs an operation based on minimum bounding rectangles. The deprecated functions have now been removed to leave only the corresponding ST_ and MBR functions:

- These functions are removed in favor of the MBR names: Contains(), Disjoint(), Equals(), Intersects(), Overlaps(), Within().
- These functions are removed in favor of the ST_names: Area(), AsBinary(), AsText(), AsWKB(), AsWKT(), Buffer(), Centroid(), ConvexHull(), Crosses(), Dimension(), Distance(), EndPoint(), Envelope(), ExteriorRing(), GeomCollFromText(), GeomCollFromWKB(), GeomFromText(), GeomFromWKB(), GeometryCollectionFromText(), GeometryCollectionFromWKB(), GeometryFromText(), GeometryFromWKB(), GeometryN(), GeometryType(), InteriorRingN(), IsClosed(), IsEmpty(), IsSimple(), LineFromText(), LineFromWKB(), LineStringFromText(), LineStringFromWKB(), MLineFromText(), MLineFromWKB(), MPointFromText(), MPointFromWKB(), MPolyFromText(), MPolyFromWKB(), MultiLineStringFromText(), MultiLineStringFromWKB(), MultiPointFromText(), MultiPointFromWKB(), MultiPolygonFromText(), MultiPolygonFromWKB(), NumGeometries(), NumInteriorRings(), NumPoints(), PointFromText(), PointFromWKB(), PointN(), PolyFromText(), PolyFromWKB(), PolygonFromText(), PolygonFromWKB(), SRID(), StartPoint(), Touches(), X(), Y().
- GLength() is removed in favor of ST_Length().

Test Suite Notes

- mysql-test-run.pl now supports a --do-suite option, which is similar to --do-test but permits specifying entire suites of tests to run. (Bug #24350345)

- The mysqltest rmdir command fails if the directory to be removed contains any files or directories. To enable recursive removal of a directory as well as its contents, if any, mysqltest now supports a force-rmdir command. (Bug #24316799)

- Two new test suite options make it easier to debug test cases:
  - mysql-test-run.pl supports a --mysqltest=option option that enables options to be passed to mysqltest.
  - mysqltest supports a --trace-exec option that causes it to immediately print output from executed programs to stdout.

mysql-test-run.pl now recognizes the MTR_CTEST_TIMEOUT environment variable. If set, the value is a timeout in seconds to pass to ctest unit test commands. (Bug #21821049, Bug #21278845)
• For test cases in the MySQL test suite, it was previously possible to use symbolic error names for the --error command only for server errors. This is now also possible for client errors. For example:

```
--error CR_SERVER_GONE_ERROR
```

(Bug #21048973, Bug #76972)

• The `mysqltest` program now has a `copy_files_wildcard` command that copies all files that match a pattern from a source directory to a destination directory. See the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html. (Bug #82111, Bug #23743035)

### X Plugin Notes

• The `Protobuf` decoder class limited the number of nested objects to 50 (the default value). (Bug #23707238, Bug #82025)

• The statement `list_objects` incorrectly reported a table as a collection. (Bug #23631240)

• The `create_collection` statement created a collection table with a unique key index on the `_id` column instead of on the primary key. (Bug #23284569)

### Functionality Added or Changed

• **Incompatible Change; Partitioning**: The generic partitioning handler has been removed from the MySQL server. As part of this change, `mysqld` no longer supports the `--partition` and `--skip-partition` options, and the server can no longer be built using `--DWITH_PARTITION_STORAGE_ENGINE`. `partition` is also no longer displayed in the output of `SHOW PLUGINS`, or shown in the `INFORMATION_SCHEMA.PLUGINS` table.

In order to support partitioning of a given table, the storage engine used for the table must now provide its own (“native”) partitioning handler. **InnoDB** is the only storage engine supported in MySQL 8.0 which includes a native partitioning handler. An attempt to create partitioned tables in MySQL 8.0 using any other storage engine fails. (The **NDB** storage engine used by MySQL NDB Cluster also provides its own partitioning handler, but is currently not supported by MySQL 8.0.)

**Effects on upgrades.** The direct upgrade of a partitioned table using a storage engine other than **InnoDB** (such as **MyISAM**) from MySQL 5.7 (or earlier) to MySQL 8.0 is not supported. There are two options for upgrading such a table to be compatible with MySQL 8.0, listed here:

• Remove the table’s partitioning; you can do this without any data loss by executing an `ALTER TABLE ... REMOVE PARTITIONING` statement.

• Change the storage engine used for the table to **InnoDB**, using `ALTER TABLE ... ENGINE=INNODB`; this leaves the table’s partitioning in place. At least one of these operations must be performed for any partitioned non-InnoDB table, prior to upgrading the server to MySQL 8.0. Otherwise, such a table cannot be used following the upgrade.

For information about converting **MyISAM** tables to **InnoDB**, see [Converting Tables from MyISAM to InnoDB](#).

An analogous situation is met when importing databases from a dump file that was created in MySQL 5.7 or earlier using `mysqldump` into a MySQL 8.0 server, due to the fact that table creation statements that would result in a partitioned table using a storage engine without such support fail with an error in MySQL 8.0. For this reason you must ensure that any statements in the dump file creating partitioned tables do not also specify an unsupported storage engine. You can do this either by removing any
references to partitioning from CREATE TABLE statements that use a value for the STORAGE ENGINE option other than InnoDB, or by specifying the storage engine as InnoDB (or allowing InnoDB to be used by default).

For more information, see Partitioning Limitations Relating to Storage Engines.

• **Important Change; InnoDB:** The following InnoDB file format configuration options were deprecated in MySQL 5.7.7 and are now removed:
  - `innodb_file_format`
  - `innodb_file_format_check`
  - `innodb_file_format_max`
  - `innodb_large_prefix`

File format configuration options were necessary for creating tables compatible with earlier versions of InnoDB in MySQL 5.1. Now that MySQL 5.1 has reached the end of its product lifecycle, these options are no longer required.

The `FILE_FORMAT` column was removed from the `INNODB_SYS_TABLES` and `INNODB_SYS_TABLESPACES` Information Schema tables.

• **InnoDB:** The `innodb_buffer_pool_debug` option permits multiple buffer pool instances when the buffer pool is less than 1GB in size, ignoring the 1GB minimum buffer pool size constraint imposed on `innodb_buffer_pool_instances`. (Bug #24287290)

• **InnoDB:** A new dynamic configuration option, `innodb_deadlock_detect`, may be used to disable deadlock detection. On high concurrency systems, deadlock detection can cause a slowdown when numerous threads wait for the same lock. At times, it may be more efficient to disable deadlock detection and rely on the `innodb_lock_wait_timeout` setting for transaction rollback when a deadlock occurs. (Bug #23477773)

• **InnoDB:** The `libinnodb_zipdecompress.a` library allows external tools to use the `page_zip_decompress_low()` function to decompress InnoDB pages. (Bug #21405300, Bug #77664)

• **InnoDB:** To address contention that could occur under some workloads, the buffer pool mutex was removed and replaced by several list and hash protecting mutexes. Also, several buffer pool related variables no longer require buffer pool mutex protection. Thanks to Yasufumi Kinoshita and Laurynas Biveinis for the patch. (Bug #20381905, Bug #75534)

• **InnoDB:** InnoDB now avoids intermediate commits that would occur every 10000 rows during `ALTER TABLE ALGORITHM=COPY` operations. The purpose of intermediate commits was to speed up recovery in the case of an aborted `ALTER TABLE ALGORITHM=COPY` operation. If an `ALTER TABLE ALGORITHM=COPY` operation is aborted, the new, uncommitted table is now dropped during DDL log recovery before the undo log is rolled back, thereby avoiding time-consuming data rollback for the uncommitted table. Undo logging is now suppressed for `ALTER TABLE ALGORITHM=COPY` operations unless there is an `IGNORE` clause or something else that requires rollback capability.

If there is full-text index on the table being altered, full-text data is inserted into full-text auxiliary tables as the `ALTER TABLE ALGORITHM=COPY` operation inserts rows into the new, uncommitted table. Previously, full-text data was only processed on transaction commit. (Bug #17479594)

• **InnoDB:** To reduce read-write lock contention that can result from multiple purge threads purging rows from the same table, undo records are now grouped and assigned to different purge threads by table ID.
MySQL 8.0 Release Notes

- **InnoDB**: InnoDB code now uses the C++ std::thread library for thread management.

- **InnoDB**: BLOB code was refactored to provide an internal C++ interface for operations on compressed and uncompressed BLOB data.

- **InnoDB**: The InnoDB memcached plugin now supports multiple get operations (fetching multiple key/value pairs in a single memcached query) and range queries. See InnoDB memcached Multiple get and Range Query Support.

- **InnoDB**: The innodb_stats_sample_pages system variable was removed. innodb_stats_sample_pages was deprecated in MySQL 5.6.3 and replaced by innodb_stats_transient_sample_pages.

- **InnoDB**: When encountering index tree corruption, InnoDB writes a corruption flag to the redo log, which makes the corruption flag crash safe. InnoDB also writes in-memory corruption flag data to an engine-private system table on each checkpoint. During recovery, InnoDB reads corruption flags from both locations and merges results before marking in-memory table and index objects as corrupt.

- **InnoDB**: The innodb_locks_unsafe_for_binlog system variable was removed. innodb_locks_unsafe_for_binlog was deprecated in MySQL 5.6.3. The READ COMMITTED isolation level provides similar functionality.

- **InnoDB**: InnoDB no longer creates .isl files (InnoDB Symbolic Link files) when creating tablespace data files outside of the MySQL data directory.

  With this change, moving a remote tablespace while the server is offline by manually modifying an .isl file is not supported.

- **InnoDB**: InnoDB no longer supports compressed temporary tables. When innodb_strict_mode is enabled (the default), CREATE TEMPORARY TABLE returns an error if ROW_FORMAT=COMPRESSED or KEY_BLOCK_SIZE is specified. If innodb_strict_mode is disabled, warnings are issued and the temporary table is created using a non-compressed row format.

  With this change, all temporary tables are created in the shared temporary tablespace, ibtmp1.

  The PER_TABLE_TABLESPACE and IS_COMPRESSED columns were removed from the INFORMATION_SCHEMA.INNODB_TEMP_TABLE_INFO.

- **InnoDB**: The new INFORMATION_SCHEMA.INNODB_CACHED_INDEXES table reports the number of index pages cached in the InnoDB buffer pool for each index.

- **InnoDB**: The innodb_checksums system variable was removed. innodb_checksums was replaced by innodb_checksum_algorithm in MySQL 5.6.3.

- **InnoDB**: InnoDB startup code was refactored.

- **InnoDB**: The innodb_flush_method default value is no longer NULL. On Unix-like systems, the default value is fsync. On Windows systems, the default value is unbuffered.

  On Windows, the innodb_flush_method setting no longer affects the innodb_use_native_aio setting. There are now two possible settings for innodb_flush_method on Windows, unbuffered (unbuffered I/O) and normal (buffered I/O). With this change, you can enable asynchronous I/O with buffered I/O, which is a new combination (innodb_use_native_aio=ON and innodb_flush_method=normal). The async_unbuffered setting was removed.

  You can now set innodb_flush_method and innodb_change_buffering configuration options using numeric values.
• **InnoDB**: The `innodb_support_xa` system variable, which enables support for two-phase commit in XA transactions, was removed. As of MySQL 5.7.10, InnoDB support for two-phase commit in XA transactions is always enabled.

• **InnoDB**: InnoDB no longer creates an `.isl` file (InnoDB Symbolic Link files) when creating a file-per-table tablespace data file outside of the MySQL data directory. InnoDB now uses the redo log to locate remote tablespace data files.

Offline relocation of a file-per-table tablespace data file created outside of the MySQL data directory by modifying the `.isl` file is no longer supported.

• **InnoDB**: The current maximum auto-increment counter value is now written to the redo log each time the value changes, and saved to an engine-private system table on each checkpoint. These changes make the current maximum auto-increment counter value persistent across server restarts. Additionally:

  - A server restart no longer cancels the effect of the `AUTO_INCREMENT = N` table option. If you initialize the auto-increment counter to a specific value, or if you alter the auto-increment counter value to a larger value, the new value is persisted across server restarts.

  - A server restart immediately following a `ROLLBACK` operation no longer results in the reuse of auto-increment values that were allocated to the rolled-back transaction.

  - If you modify an `AUTO_INCREMENT` column value to a value larger than the current maximum auto-increment value (in an `UPDATE` operation, for example), the new value is persisted, and subsequent `INSERT` operations allocate auto-increment values starting from the new, larger value.

For more information, see [AUTO_INCREMENT Handling in InnoDB](https://mariadb.com/kb/en/mariadb/auto-increment-handling-in-innodb/) and [InnoDB AUTO_INCREMENT Counter Initialization](https://mariadb.com/kb/en/mariadb/innoDB-auto-increment-counter-initialization/). (Bug #199, Bug #13726455)

• **Replication**: There are two improvements to how a `CHANGE MASTER TO` statement is written into the error log (`mysqlld.log`):

  - Before, no commas were put between the option specifications (for example `MASTER_USER =` and `MASTER_PASSWORD =`), so users who wanted to use the statement by copy and paste had to insert the commas manually. Commas are now inserted when the statement is written to the error log.

  - When the literal “<secret>” is inserted as a placeholder for the `MASTER_PASSWORD` value, no quotes are used now, so users who forget to replace the literal with the real password before a copy and paste gets a syntax error immediately, instead of running into other issues.

(Bug #18194384)

• **Replication**: It is now possible to restore a backup of a GTID-based replication server because you can add GTIDs to `gtid_purged`, regardless of whether `gtid_executed` is empty or not. This enables you to restore backups from GTID-based replication servers without losing existing GTID information and binary logs. The GTIDs to add are those which existed in `gtid_executed` at the time of taking the backup. The syntax for `SET GTID_PURGED` has been extended so that `SET GTID_PURGED ="+gtid_set"` adds `gtid_set` to the existing `gtid_purged` GTID set.

• **Replication**: New Performance Schema stages have been added to show the progress of row-based replication. You can use these stages to check the progress of slow operations in row-based replication. Additionally you can find out which database the changes are being applied to. This assists in troubleshooting row-based replication issues and provides more information for performance tuning. For more information see [Monitoring Row-based Replication](https://mariadb.com/kb/en/mariadb/monitoring-row-based-replication/)

• **JSON**: This release adds an unquoting extraction operator `-->`, sometimes also referred to as an inline path operator, for use with JSON documents stored in MySQL. The new operator is similar to the `->`
operator, but performs JSON unquoting of the value as well. For a JSON column `mycol` and JSON path expression `mypath`, the following three expressions are equivalent:

- `JSON_UNQUOTE(JSON_EXTRACT(mycol, "$.mypath"))`
- `JSON_UNQUOTE(mycol->"$.mypath")`
- `mycol->>"$.mypath"

The `->>` operator can be used in SQL statements wherever `JSON_UNQUOTE(JSON_EXTRACT())` would be allowed. This includes (but is not limited to) `SELECT` lists, `WHERE` and `HAVING` clauses, and `ORDER BY` and `GROUP BY` clauses.

For more information, see Functions That Search JSON Values, and JSON Path Syntax. (Bug #78736, Bug #21980346)

- To produce more accurate estimates, the MEMORY storage engine now calculates index statistics (records per key estimates) using floating-point rather than integer arithmetic. (Bug #23024059)

- A new CMake option, INSTALL_STATIC_LIBRARIES, enables control over whether to install static libraries. The default is ON. If set to OFF, these libraries are not installed: `libmysqlclient.a`, `libmysqld.a`, `libmysqldservices.a`. (Bug #22891432)

- The internal `mysql_prepare_create_table()` server function has been refactored for improved code maintainability and clarity. This code revision results in the following minor changes of behavior for `CREATE TABLE` and `ALTER TABLE`:

  - Attempts to create a second primary key based on a NULL column now produce an `ER_MULTIPLE_PRI_KEY` error rather than `ER_PRIMARY_CANT_HAVE_NULL`.
  - Attempts to create a second primary key based on a generated column now produce an `ER_MULTIPLE_PRI_KEY` error rather than `ER_UNSUPPORTED_ACTION_ON_GENERATED_COLUMN`.
  - Attempts to create a full-text key on a JSON column now produce an `ER_JSON_USED_AS_KEY` error rather than `ER_BAD_FT_COLUMN`.
  - Attempts to create a key in a storage engine that does not support keys (for example, EXAMPLE) now produce an `ER_TOO_MANY_KEYS` error rather than `ER_TOO_MANY_KEY_PARTS`.

(Bug #22884886)

- Previously, the Performance Schema was not built for `libmysqld`, the embedded server. This prevented use of the `SHOW STATUS` and `SHOW VARIABLES` statements with `show_compatibility_56=OFF` because, with that setting, those statements take their results from Performance Schema tables. Now for `libmysqld`, the required Performance Schema tables are built (with no instrumentation collected), so that those `SHOW` statements can be supported with `show_compatibility_56=OFF`. (Bug #22809694)

- Several internal functions used by `JSON_CONTAINS()`, `JSON_SEARCH()`, and other MySQL JSON functions created excessive numbers of local copies of keys, values, or both, when performing inspections of JSON objects. Such copying has been eliminated or reduced in many cases. In addition, the lifetimes of temporary objects used by some of these functions have been reduced. These changes should make these and related JSON functions perform more efficiently than previously, and with fewer resources required. (Bug #22602142)

- If the system `lz4` and `openssl` `zlib` commands are available, the `lz4_decompress` and `zlib_decompress` utilities are unneeded. Two changes enable those utilities not to be built: If the new
WITH_LZ4 CMake option is set to system, lz4_decompress is not built or installed. If the WITH_ZLIB CMake option is set to system, zlib_decompress is not built or installed. (Bug #22329851)

- Source files for the MySQL strings library have been converted from C (.c suffix) to C++ (.cc suffix). This enables stricter compilation checks and use of C++ features in the library code. (Bug #22124719)

- Source code for the msys library now uses C++ rather than C to take advantage of stricter compilation checks and permit use of C++ features. (Bug #21881278)

- For improved performance and better compatibility with other platforms, my_sync() on OS X now uses fsync() rather than fcntl() with the F_FULLSYNC flag. (Bug #20742269)

- A new CMake option, WITH_TSAN, permits enabling ThreadSanitizer for compilers that support it. (Bug #80409, Bug #23171902)

- The global list of connections, previously protected by a single mutex, has been partitioned into eight parts, each protected by its own instance of the mutex. The result is a reduction of overhead and improved performance for connection processing. An implication of this change for monitoring purposes is that the Performance Schema now exposes eight different instances each of the LOCK_thd_list mutex, LOCK_thd_remove mutex, and COND_thd_list condition variable.

- MySQL now provides functions to manipulate UUID values and make them easier to work with:
  - UUID_TO_BIN() and BIN_TO_UUID() convert between UUID values in string and binary formats (represented as hexadecimal characters and VARBINARY(16), respectively). This permits conversion of string UUID values to binary values that take less storage space. UUID values converted to binary can be represented in a way that permits improved indexing efficiency.
  - IS_UUID() returns 1 or 0 to indicate whether its argument is a valid string-format UUID value.

For more information about these functions, see Miscellaneous Functions

- The mysql_plugin utility has been removed. Alternatives include loading plugins at server startup using the --plugin-load or --plugin-load-add option, or at runtime using the INSTALL PLUGIN statement.

- The server now relies on storage engines to clean up temporary tables left from previous server runs. InnoDB does this by discarding the temporary tablespace on restart. MyISAM and other similar storage engines still rely on scanning the temporary directory to detect leftover tables, by looking for files belonging to these engines with a certain name pattern.

- The deprecated mysql_shutdown() C API function and corresponding COM_SHUTDOWN client/server protocol command have been removed. Instead, use mysql_query() to execute a SHUTDOWN statement.

- The server no longer performs conversion of pre-MySQL 5.1 database names containing special characters to 5.1 format with the addition of a #mysql50# prefix. Because these conversions are no longer performed, the --fix-db-names and --fix-table-names options for mysqlcheck, the UPGRADE DATA DIRECTORY NAME clause for the ALTER DATABASE statement, and the Com_alter_db_upgrade status variable have been removed.

  Upgrades are supported only from one major version to another (for example, 5.0 to 5.1, or 5.1 to 5.5), so there should be little remaining need for conversion of older 5.0 database names to current versions of MySQL. As a workaround, upgrade a MySQL 5.0 installation to MySQL 5.1 before upgrading to a more recent release.
Bugs Fixed

- **Incompatible Change**: Concatenation of spatial values makes little sense, so the `CONCAT()` and `CONCAT_WS()` functions now produce an error for spatial arguments. (Bug #22893669)

- **Important Change; JSON**: The empty string value is now accepted as a key when used with `JSON` functions such as `JSON_EXTRACT()`. In such cases, it must be quoted. (Bug #79643, Bug #22366102)

- **NDB Cluster**: Previously, the `mysql.ndb_binlog_index` table was created even if the server was built without `NDB`. Now the table is created only if the server is built with `NDB`. (Bug #22874872)

- **InnoDB; Microsoft Windows**: An unspecified block size resulted in an empty `INFORMATION_SCHEMA.TABLESPACES` table on Windows NTFS with a cluster page size greater than or equal to 8K. (Bug #23598872)

- **InnoDB; Microsoft Windows**: Compilation of InnoDB with Visual Studio 2015 Update 2 returned warnings. (Bug #23056963)

- **InnoDB**: An ALTER TABLE ... ALGORITHM=COPY operation that added a foreign key constraint failed after due to an intermediate commit that occurred after 10000 rows were copied. The intermediate commit reset the foreign key checks flag, causing the operation to fail. The intermediate commit no longer occurs. (Bug #28662255, Bug #92471)

  References: See also: Bug #17479594.

- **InnoDB**: Unnecessary checks were removed from the `ut_cpuid()` function which is used to fetch information about the CPU. (Bug #24405292)

- **InnoDB**: An asynchronous read operation on a deleted tablespace raised an error. (Bug #24388498)

- **InnoDB**: `dict_col_t` accessors were added to the InnoDB code. (Bug #24363566)

- **InnoDB**: `dict_col` functions in the InnoDB code were replaced by accessors. (Bug #24361098)

- **InnoDB**: `dict_index_t` functions in the InnoDB code were replaced by accessors. (Bug #24361023)

- **InnoDB**: Unnecessary code that checked for and released reserved adaptive hash index search latches was removed. (Bug #24300175)

- **InnoDB**: A system tablespace data file size greater than 4G on a 32-bit operating system could result in an overflow condition. (Bug #23753625)

- **InnoDB**: Internal methods for accessing table object data were added to `dict_table_t`. (Bug #23748128)

- **InnoDB**: The restriction that required the first undo tablespace to use `space_id` 1 was removed to avoid `space_id` conflicts with existing tablespaces during upgrade. The first undo tablespace can now use a `space_id` other than 1. `space_id` values for undo tablespaces are still assigned in a consecutive sequence. (Bug #23517560)

- **InnoDB**: Internal accessor functions for iterating the indexes of a table were replaced with accessor methods. Dead code was removed. (Bug #23336108)

- **InnoDB**: The `mysql.innodb_index_stats` and `mysql.innodb_table_stats` table definitions, which were previously created by an SQL script, are now hard-coded. As a result, the `dict_table_schema_check` function is longer required and was removed. (Bug #23336079)

- **InnoDB**: The `ut_snprint` function was replaced by the C++11 `snprintf` function. (Bug #23329353)
• **InnoDB**: For consistency, instances of `ulint` in InnoDB code were replaced with `space_id_t` and `page_no_t` data types. (Bug #23297169)

• **InnoDB**: Use of boost::atomic in InnoDB code was replaced with std::atomic. (Bug #23280649)

• **InnoDB**: MySQL binaries were not built with the NUMA feature. (Bug #23259754)

• **InnoDB**: References to `UNIV_NONINL` and `UNIV_MUST_NOT_INLINE` were removed. The `fut0fut.cc` and `ut0byte.cc` files, which were only necessary when `UNIV_NONINL` was defined, were also removed. (Bug #23150562)

• **InnoDB**: The `mutex_own()` mapping caused warnings when compiling with Clang or newer GCC compilers. (Bug #23090278)

• **InnoDB**: Querying the Performance Schema for InnoDB memory allocation event data incorrectly reported values of 0. (Bug #23020280)

• **InnoDB**: `DBG_OFF` compile-time flags were replaced by `UNIV_DEBUG` flags. To improve error log output, `ut_dbg_assertion_failed()` now uses `sql_print_error()` to display the file name, line number, and message in a single line. The thread ID is displayed in a subsequent line. (Bug #22996442, Bug #23028144)

• **InnoDB**: `SHOW ENGINE INNODB STATUS` output displayed negative spin rounds per wait values. Thanks to Laurynas Biveinis for the patch. (Bug #22844987, Bug #79703)

• **InnoDB**: The `innodb_disable_resize_buffer_pool_debug` option was removed. The patch for this change also removed a code variable and simplified the `buf_pool_resize()` function. (Bug #22755053)

• **InnoDB**: After a successful `ALTER TABLE ... ALGORITHM=COPY` operation, an assertion was raised while building a previous version of a clustered index record. (Bug #22707367)

• **InnoDB**: A DML operation that updated a counter in a table with a virtual index raised on assertion in `row_parse_int()`. (Bug #22650195)

• **InnoDB**: The InnoDB memcached plugin would not load when compiled with libevent 2.0. (Bug #22646919)

• **InnoDB**: Unused calculations for integer-based `rec_per_key` values were removed from InnoDB. Integer-based `rec_per_key` information was replaced by floating point index statistics in an earlier release. (Bug #22623548)

• **InnoDB**: On slow shutdown, purge thread shutdown was initiated before the background rollback thread exited, resulting in an assertion failure. (Bug #22561332)

• **InnoDB**: Blocks were lost in `row_vers_old_has_index_entry()` due to unfreed heaps. (Bug #22543834, Bug #79973)

• **InnoDB**: A transportable tablespace debug test raised an assertion that was due to a race condition. (Bug #22453668)

• **InnoDB**: The server failed to start due to missing undo tablespaces. (Bug #22452992)

• **InnoDB**: InnoDB recovery asserted while attempting to close an undo tablespace due to buffered undo tablespace changes introduced by the recovery process. (Bug #22361764)

• **InnoDB**: Reallocation of memcached-referenced memory raised an assertion. (Bug #22304250, Bug #79500)
MySQL 8.0 Release Notes

• **InnoDB**: Building InnoDB with C++11 returned "register" deprecation warnings. Handling of "register" deprecation warnings remained in the code after the deprecated "register" keyword was removed. Also, an unused declaration of `yyset_extra()` was removed. (Bug #22292704)

• **InnoDB**: `SHOW CREATE TABLE` output for partitioned tables did not accurately display tablespace assignment information for table partitions. (Bug #22245554)

• **InnoDB**: An `ALTER TABLE...TRUNCATE PARTITION` operation ignored the table's `KEY_BLOCK_SIZE` attribute and used the default value instead, which is half of the `innodb_page_size` value. (Bug #22186558, Bug #79223)

• **InnoDB**: Memory leaks in `innochecksum` were corrected. (Bug #22179518)

• **InnoDB**: A `SPACE_ID` column was added to the `INNODB_CACHED_INDEXES` table. The `INDEX_ID` value is no longer a global unique identifier. (Bug #22172026)

• **InnoDB**: A purge thread open table callback for virtual columns raised an assertion due to an unexpected data dictionary table latch. As a temporary workaround, purge is temporarily disabled for virtual generated columns. This temporary workaround may cause b-tree expansion due to unpurged delete-marked records for indexes on virtual columns. (Bug #22153217)

• **InnoDB**: Creating a table with a full-text index and a foreign key constraint failed when `foreign_key_checks` was disabled. (Bug #22094601, Bug #78955)

References: This issue is a regression of: Bug #16845421.

• **InnoDB**: The `ha_innobase::m_primary_key` field was removed. It was redundant. A boolean predicate, `TABLE_SHARE::is_missing_primary_key()`, was added. (Bug #21928734, Bug #78662)

• **InnoDB**: A buffer pool load operation that attempted to load an uninitialized page caused a Valgrind failure. (Bug #21747906)

• **InnoDB**: Unused functions introduced by the `InnoDB memcached` plugin were removed. (Bug #21625760)

• **InnoDB**: An internal global variable used by the `innodb_buffer_pool_size` configuration option was removed. (Bug #21512749)

• **InnoDB**: An `InnoDB` page cleaner thread asserted during a buffer pool resize operation. (Bug #21473497)

• **InnoDB**: Log buffer contention was reduced with the addition of a second buffer, allowing for concurrent log buffer writing and flushing. A new mutex was added to protect log buffer flushing. Thanks to Zhai Weixiang for the patch. (Bug #21352937, Bug #77094)

• **InnoDB**: Unused `InnoDB` and `libsql` functions and variables were removed, and global symbols were converted to static keywords, where possible. (Bug #21153166, Bug #21141390, Bug #77146, Bug #21178589)

• **InnoDB**: A number of unused predefined functions were removed from an internal SQL parser that is used for implementing full-text indexes, updating the data dictionary, and updating persistent statistics. (Bug #21126390, Bug #77111)

• **InnoDB**: The `SysTablespace::parse_units()` function now returns the number of pages in a file instead of the number of megabytes. The `SysTablespace::normalize_size()` function was removed. Error messages in `SysTablespace::parse_params()` were revised. (Bug #21040199, Bug #76949)
• **InnoDB**: For persistent tables, the internal unique identifier for InnoDB indexes (`index_id`) now includes a tablespace identifier (`space_id`), making index identifiers unique at the tablespace level as well as the InnoDB instance level, and supports future work related to index identifier allocation. (Bug #20737524, Bug #76392)

• **InnoDB**: In `mtr0mtr.cc`, a redundant function was removed, and the `ReleaseBlocks` function was renamed to `AddDirtyBlocksToFlushList`. (Bug #20735882, Bug #76343)

• **InnoDB**: Code related to `innochecksum` was cleaned up and reorganized. Checksum functionality is now located in `buf0checksum.cc`. (Bug #20518099)

• **InnoDB**: `__attribute__((nonnull))` was removed from InnoDB code. The attribute is no longer permitted by InnoDB coding guidelines. (Bug #20468234)

• **InnoDB**: A new struct was added to provide a logical interface for handling and manipulating external BLOB field references. (Bug #18195972)

• **InnoDB**: `TRUNCATE TABLE` is now mapped to `DROP TABLE` and `CREATE TABLE`. This change has the following implications:
  
  - On systems with a large buffer pool and `innodb_adaptive_hash_index` enabled, `TRUNCATE TABLE` operations previously caused a temporary drop in system performance due to an LRU scan that occurred when removing the table’s adaptive hash index entries. The remapping of `TRUNCATE TABLE` to `DROP TABLE` and `CREATE TABLE` avoids the problematic LRU scan.
  
  - `TRUNCATE TABLE` is temporarily non-atomic. A server exit during a `TRUNCATE TABLE` operation can result in a dropped table and orphaned foreign key constraints in the InnoDB `SYS_FOREIGN` and `SYS_FOREIGN_COLS` system tables.
  
  - The InnoDB memcached plugin `flush_all` command invokes `DELETE` instead of `TRUNCATE TABLE`. `DELETE` has a higher overhead cost than `FLUSH TABLES` since it involves undo-logging, delete-marking, and eventually purging each deleted row.

  - A log checkpoint that occurred for internal truncate table operations on file-per-table tablespaces was replaced by a log flush.

  (Bug #16834993, Bug #68184, Bug #16207919)

• **InnoDB**: InnoDB wasted most pages in extents used for fragment pages. (Bug #16204823, Bug #67963)

• **Partitioning**: In some cases, an issue with partition pruning being attempted a second time during optimization after all partitions had already been pruned at parsing time led to an assert. (Bug #23194259)

• **Partitioning**: A partitioned table whose table name and any partition name had a combined length in excess of 61 characters could not be imported from a backup created using `mysqldump`. When the table also employed subpartitioning, then the combined length of the table name, any partition name, and the name of any subpartition of this partition could not exceed 57 characters without triggering the same issue.

  This was due to the fact that the internal `mysql.innodb_table_stats` table allowed a maximum of 64 characters for the column used to store the table name, even though InnoDB stores, for a partitioned or subpartitioned table, a row in `innodb_table_stats` for each partition or subpartition wherein the value actually used to represent the table name follows the pattern `table_name#P#partition_name` or `table_name#P#partition_name#SP#subpartition_name`, respectively. This issue is fixed by
changing the definition of the `innodb_table_stats` to accommodate the maximum combined length of these attributes plus `#P#` and `#SP#` (199 characters). (Bug #72061, Bug #18416479)

- **Replication:** In `Slave_worker::write_info()`, `DEBUG_ENTER()` had "Master_info::write_info" as its argument instead of "Slave_worker::write_info". This fix corrects the argument. Thanks to Stewart Smith for the patch. (Bug #21658067, Bug #78133)

- **Replication:** When using `START SLAVE UNTIL position` statements with a multithreaded slave the only `UNTIL` clause available was `SQL_BEFORE_GTIDS`. Now multithreaded slaves are compatible with the `START SLAVE UNTIL MASTER_LOG_FILE = 'log_name', MASTER_LOG_POS = log_position` and `START SLAVE UNTIL RELAY_LOG_FILE = 'log_name', RELAY_LOG_POS = log_position` statements. (Bug #75843, Bug #20513547)

- **Microsoft Windows:** On Windows, setting the global `log_syslog` system variable in multiple threads could cause a server exit. (Bug #22180046)

- **Solaris:** The client library failed to build on Solaris using the `Cstd` library. (Bug #24353920, Bug #82347)

- **JSON:** `CHECKSUM_TABLE` calculated the checksums for `JSON` values using the memory addresses of the values rather than the values themselves, which made the checksum vary. Now in such cases the calculation is based on the actual `JSON` value, and not on that value’s address. (Bug #23535703)

- **JSON:** Passing `NULL` to a stored procedure expecting a `JSON` parameter led to an assertion failure in debug builds. (Bug #23209914)

- **JSON:** Parsing of `JSON` path arguments failed to distinguish between a NULL path and one that was syntactically invalid.

  This has been changed so that parsing of these paths now clearly distinguishes between valid non-NUL paths, NULL paths, and invalid paths. (Bug #22816576)

- **JSON:** For debug builds, an assertion could be raised when the server created a temporary table to hold `JSON` objects. (Bug #22782948)

- **JSON:** Queries that executed a `JSON` function that raised an error could cause a server exit. (Bug #22253965)

- Renaming a table to be part of a nonexistent database failed (correctly), but with an Unknown error message. A proper error message is now produced; this was corrected as part of the data dictionary implementation. (Bug #25167507, Bug #84000)

- For segmentation faults on FreeBSD, the server did not generate a stack trace. (Bug #24566529, Bug #23575445, Bug #81827)

- On macOS, stack trace demangling now occurs for builds compiled using Clang, just as for GCC. (Bug #23606094, Bug #81908)

- `libevent` was built on macOS even when not needed. (Bug #23228287, Bug #81311)

- A function that returns a `JSON` value could cause a server exit if called as part of a `CASE` statement in a stored procedure. (Bug #23212765)

- Previously, different values were reported by `SHOW ENGINE PERFORMANCE_SCHEMA STATUS` and `SELECT * FROM performance_schema.memory_summary_global_by_event_name` for total memory used in the Performance Schema. The memory for scalable buffer pages, instrumented as `memory/performance_schema/scalable_buffer`, was missing from the
SHOW ENGINE STATUS output. That statement now includes the missing memory, displayed as (pfs_buffer_scalable_container).memory. (Bug #23104498)

• The -fexpensive-optimizations option to GCC caused ARM64 and PowerPC builds to compute floating-point operations slightly differently from other platforms. This option was enabled by -O2 and higher optimization levels. The option now is disabled on platforms negatively affected by it. (Bug #23046775)

• After a failed administrative operation such as ALTER TABLE ... OPTIMIZE PARTITION, selecting from the Performance Schema in lock-tables mode could hang. (Bug #23044286)

• In builds with AddressSanitizer enabled, CAST(... AS BINARY) could cause a server exit. (Bug #22900560)

• Some spatial functions were reported using a different spatial function name in error messages. (Bug #22883056, Bug #80627)

• Fixed Valgrind warnings with Clang in optimized mode for the my_strtod_int() function. (Bug #22839888)

• Precision math operations on values with 64 decimals could produce a 0 result. (Bug #22828692)

• ST_GeomFromGeoJSON() could return an error with valid arguments. (Bug #22804853)

• For debug builds with STRICT_TRANS_TABLES SQL mode enabled, an assertion could be raised by INSERT or REPLACE statements that had made changes that could not be rolled back and that subsequently generated an ER_NO_DEFAULT_FOR_FIELD error. (Bug #22635253)

• The optimizer contained a memcpy() call that did not check for overlapping source and destination. (Bug #22537196)

• For debug builds, a missing error check on the result of a subquery that accessed a JSON value could raise an assertion. (Bug #22522073)

• Preparing a CREATE TABLE ... SELECT statement, then flushing tables (thus closing the table) and executing the prepared statement could cause a server exit. (Bug #22393309)

• A prepared statement that used a parameter in the select list of a derived table that was part of a join could cause a server exit. (Bug #22392374, Bug #24380263)

• Some grant tables did not account for the increase in maximum user name length from 16 to 32 characters in MySQL 5.7.8. (Bug #22379607, Bug #79680)

• Re-evaluation of a generated column expression could cause access to previously freed memory and a server exit. (Bug #22346120)

• HANDLER read statements that searched an index when the target index value was not stored into the row buffer successfully could cause a server exit. (Bug #22321965)

• Improper handling of numeric-to-ZEROFILL conversion for NULL values could lead to a server exit. (Bug #22281205)

• Using a subquery containing a row constructor to set a variable in a SET statement could cause a server exit. (Bug #22276843)

• If the SQL mode did not include ALLOW_INVALID_DATES, a query that contained invalid_date IN (subquery) and was handled by subquery materialization could cause a server exit. (Bug #22262843)
• For the embedded server, the code following the check for invalid arguments was invoked with missing or incorrect arguments, which could lead to an improper exit. (Bug #22262706)

• On OS X, `vio_io_wait()` used `select()`, limiting the number of file descriptors to 1024. Now kqueue event notification is used instead to avoid this limit. FreeBSD was changed to use kqueue as well. (Bug #22244911)

• Memory leaks could result if stored routine loading involved temporarily changing the default database and errors occurred restoring the original default database. (Bug #22179795)

• CMake configuration was adjusted to check for `-Wxxx` compiler options instead of `-Wno-xxx` because the latter produce false positives for GCC. (Bug #21881753)

• There could be discrepancies between the values of `INFORMATION_SCHEMA.EVENTS.LAST_EXECUTED` and `mysql.event.last_executed`. This no longer occurs. Event information is stored in the `mysql.events` data dictionary table, which is invisible, so that `INFORMATION_SCHEMA.EVENTS` is the sole interface to event metadata. (Bug #21374010)

• Views could evaluate user-defined or SQL functions before evaluating restrictions from the view definition. (Bug #20933307)

• With `-DENABLE_DTRACE=ON`, CMake did not check whether a working DTrace installation was present. Now it checks and aborts if DTrace cannot be found. (Bug #20671056)

• If given a relative path name for the `--log-error` option, `mysqld` could send `stdout` and `stderr` to the wrong location. (Bug #20609063)

• Evaluation of `LEAST()` and `GREATEST()` could use too small a sort buffer for datetime and string literals, causing an assertion to be raised. (Bug #20565160)

• The range of error numbers for errors that are new in MySQL 8.0 has been designated to begin with 3500. (Bug #20538173)

• Debian packaging was updated not to set the `sql_mode` system variable in `my.cnf`. (Bug #20535729)

• Event loading from the `mysql.event` system table could fail if the `PAD_CHAR_TO_FULL_LENGTH` SQL mode was enabled. (Bug #20073523, Bug #74947)

• Statements such as `INSERT` and `LOAD DATA` that use the `REPLACE` or `IGNORE` keyword to handle duplicate records could affect subsequent operations. (Bug #20017428)

• `CREATE TABLE ... SELECT` where non-BIT data was selected from the source table into a BIT column in the destination table could cause a server exit. (Bug #19930894)

• Compilation failed on OS X when MySQL was configured with `-DMYSQL_MAINTAINER_MODE=1` and compiled with clang/Xcode 6.0. (Bug #19694515, Bug #74100)

• For `CHANGE MASTER TO` statements rewritten to filter the password before being written to the general query log, any `MASTER_AUTO_POSITION` clause was lost. (Bug #19622609)

• Use of the `VALUES()` function in a `SELECT` clause could result in a server exit. (Bug #19601973)

• A potential null-pointer dereference and memory leak in table-rename code were corrected. (Bug #18194270)

• Using `MATCH ... AGAINST` to compare a character column and an aggregate function could cause a server exit. (Bug #17865492)
MySQL 8.0 Release Notes

• Slightly different values for the number of connections could be reported in various information sources, such as the `Connections` status variable, Performance Schema `threads` and `global_status` tables, and `SHOW PROCESSLIST` statement. (Bug #17666696)

• A query with a subquery containing a set operation with an outer reference might cause a server exit. (Bug #17270896)

• Using `GRANT` to change a password for an invalid user produced an error, but also updated the `mysql.user` system table. (Bug #17180985)

• The parser for spatial WKT data accepted numbers such as 0.23 but not .23, the equivalent value without the leading zero. Now both formats are accepted. (Bug #17167633)

• Previously, if a client attempted to send connection attribute key/value pairs that in aggregate had a size larger than the value of the `performance_schema_session_connect_attrs_size` system variable, the Performance Schema truncated the attribute data. In addition, the Performance Schema wrote this message to the error log if the `log_warnings` system variable was greater than zero:

  `[Warning] Connection attributes of length N were truncated`

This message was not helpful to a DBA attempting to determine the problematic client, so several changes have been made to connection attribute handling:

• Truncation of connection attributes still occurs for excessive data, but the log message is more informative. It includes the number of bytes lost, the connection identifier, and information about the client user. The additional information should enable DBAs to more easily identify clients for which attribute truncation occurred.

• When truncation occurs, a `_truncated` attribute is added to the session attributes with a value indicating how many bytes were lost, if the attribute buffer has sufficient space. This enables the Performance Schema to expose per-connection truncation information in the connection attribute tables.

• A new status variable, `performance_schema_session_connect_attrs_longest_seen`, indicates the longest connection attribute buffer smaller than 64KB seen by the server. If this value is larger than `performance_schema_session_connect_attrs_size`, attribute truncation has occurred, and DBAs may wish to increase the latter value, or, alternatively, investigate which clients are sending large amounts of attribute data.

For more information, see Performance Schema Connection Attribute Tables. (Bug #16576959)

• Introduction of the data dictionary enabled several `INFORMATION_SCHEMA8e` problems to be addressed:

  • Queries on `INFORMATION_SCHEMA.STATISTICS` could return different results depending on the order of columns in the select list.

  • Some `INFORMATION_SCHEMA` tables had suboptimal column types and sizes. Such tables that are now views on data dictionary tables in the `mysql` system database have more appropriate column definitions.

  • Queries on `INFORMATION_SCHEMA` tables that resulted in directory scans to determine database or file names no longer do so, but instead read database and table names from the data dictionary.

  • Queries on `INFORMATION_SCHEMA` tables that opened `.frm` files to obtain table metadata no longer do so, but instead read this information from the data dictionary.
• For comparisons of database or table names in INFORMATION_SCHEMA queries, using COLLATE to force a given collation worked only if applied to the INFORMATION_SCHEMA table column, but not if applied to the comparison value.

For additional information about performing such comparisons, see Using Collation in INFORMATION_SCHEMA Searches.

(Bug #14017351, Bug #65121, Bug #17559183, Bug #70462, Bug #23259470, Bug #81347, Bug #20372562, Bug #75532, Bug #13878164, Bug #11756519, Bug #48445)

• For abnormal server exit on Windows, the server previously created a minidump file named module_name.dmp, where module_name is the name of the server executable file. To prevent earlier minidump files from being overwritten, minidump file names now include the process ID and have the form module_name..pid.dmp; for example, mysql.exe.7296.dmp. (Bug #12779463)

• For queries on INFORMATION_SCHEMA tables, comparisons of schema and table names could be case sensitive or insensitive, depending on the characteristics of the underlying file system and the lower_case_table_names system variable value. Furthermore, it was ineffective to provide a COLLATE clause to change the comparison properties because that clause was ignored. This has been changed so that COLLATE is no longer ignored and can be used to obtain the desired comparison properties. (Bug #11748044, Bug #34921)

• FLOOR(CEIL()) truncated large BIGINT UNSIGNED arguments. (Bug #80873, Bug #23013359)

• Manipulation of a value returned by the JSON_MERGE() function using JSON_SET() sometimes produced an invalid result. (Bug #80787, Bug #22961128)

• ST_AsGeoJSON() failed when geometry arguments were supplied using user-defined variables. (Bug #80697, Bug #22912800)

• CAST(expr AS BINARY(N)) unexpectedly returned NULL for some valid values of N. (Bug #80630, Bug #22885819)

• Geometry import functions that took an SRID parameter cast it to an unsigned 32-bit integer without warning or error, so negative values or values larger than unsigned 32-bit integer range were silently converted to a number within the range. Now, all geometry functions that take the SRID as a parameter check that it is within unsigned 32-bit integer range and produce an ER_DATA_OUT_OF_RANGE error if not. This also applies to GeoJSON and GeoHash functions that previously checked that the parameter was within range but returned a different error code. (Bug #80499, Bug #22819614)

• If rounding occurred while storing a predicate value, the range optimizer might not return correct results for the < and <= operators. (Bug #80244, Bug #22661012)

• For the mf_ioctl test, add a missing va_end(), fix a memory leak by calling my_end(), and add a target for the test. Thanks to Daniel Black for the patch on which these changes were based. (Bug #80085, Bug #22578670)

• SELECT DISTINCT SUBSTR() could incorrectly discard values as duplicates for large position or length arguments. The same issue also affected LEFT() and RIGHT(). (Bug #80047, Bug #22565155)

• SUBSTRING_INDEX(str, delim, count) did not properly handle count values larger than 32 bits. (Bug #79978, Bug #22545429)

• For calls to CONVERT(), literal string arguments could be modified during execution, producing incorrect results. (Bug #79924, Bug #22531111)

• REPLACE('a', BINARY 'b', NULL) returned 'a' rather than NULL. (Bug #79912, Bug #22523836)
• Transaction state tracking now avoids a function that is not 8-bit safe, for enhanced compatibility with nonstandard character sets. (Bug #79905, Bug #22523383)

• GREATEST() and LEAST() treated all integer input as signed. (Bug #79902, Bug #22523685)

• Lines in the general query log were missing a tab between the timestamp and the thread ID. Thanks to Tsubasa Tanaka for the patch. (Bug #79868, Bug #22508563)

• Some comparisons between unsigned values and negative upper limits could return incorrect results (for example, CAST(100 AS UNSIGNED) BETWEEN 1 AND -1). (Bug #79857, Bug #22501606)

• The REPEAT() function did not properly handle output from the SUBSTR() function. (Bug #79695, Bug #22391186)

• The JSON_TYPE() function now shows the type of BIT literals cast to JSON as BLOB, rather than BIT. (Bug #79308, Bug #22297987)

• Configuring MySQL with the -DWITH_UBSAN=ON CMake option produced a server that was not fully functional. (Bug #79238, Bug #22194071)

• sql_common.h, a header file included in MySQL distributions, included and was therefore dependent on hash.h, a header file not included in MySQL distributions. This resulted in compilation failures. To eliminate this dependency, sql_common.h was modified to no longer include hash.h. (Bug #79237, Bug #70672, Bug #17633467)

• Timers used for checking maximum statement execution time were initialized even when the server was started with the --help option. If --help is given, this is no longer done. (Bug #79182, Bug #22172389)

• The optimizer failed when trying to optimize away expressions of the form IF(true, '2015-01-01', '2015-01-01') IS NOT NULL. (Bug #79114, Bug #22148586)

• Subtraction of an unsigned decimal could return a negative value, but with metadata type information of UNSIGNED BINARY. Subtraction for unsigned decimal subtraction now is handled the same way as for unsigned integer: Produce an ER_DATA_OUT_OF_RANGE error if the result is negative, unless the NO_UNSIGNED_SUBTRACTION SQL mode is enabled. (Bug #78914, Bug #22083757)

• Handling by the HEX() function of numbers larger than $2^{64}$ was improved. (Bug #78828, Bug #22297983)

References: This issue is a regression of: Bug #9854.

• The client-side plugin deinitialization function signature was changed from int (*deinit)() to int (*deinit)(void) to avoid warnings when compiling with -Wstrict-prototypes. (Bug #78177, Bug #21680094, Bug #81419, Bug #23282498)

• CREATE TABLE reported an incorrect error if a very long or incorrect path name was specified for the DATA DIRECTORY or INDEX DIRECTORY table option. Now ER_PATH_LENGTH or ER_WRONG_VALUE are reported for those cases. (Bug #76635, Bug #20857556)

• The server now tries to provide more informative messages for these error codes: ER_CANT_CREATE_DB, ER_CANT_CREATE_TABLE, ER_DB_DROP_DELETE, ER_DB_DROP_RMDIR, ER_ERROR_DURING_COMMIT, ER_ERROR_DURING_ROLLBACK, ER_GET_ERRNO. (Bug #76298, Bug #20694494)

• For some instances of failure to prepare an XA transaction, incomplete transaction cleanup could raise an assertion. (Bug #75809, Bug #20488921)
• **mysql** could attempt to close an invalid socket file descriptor. Thanks to Zhai Weixiang for the patch. (Bug #75778, Bug #20504513)

• A statement of the following form converted the table data to **latin1**, but also changed the table default character set to **latin1** and ignored the **utf8** clause:

```
ALTER TABLE tbl_name CHARACTER SET utf8, CONVERT TO CHARACTER SET latin1;
```

Thanks to Daniel Black for the patch. (Bug #75320, Bug #20279241)

• In **mysql.cc**, the **abort_loop** variable was quantified with **volatile**, which on some platforms could result in changes not being seen immediately in threads running on different cores. Thanks to Stewart Smith for the patch. (Bug #74846, Bug #20134637)

• Calling a procedure which created a view from a trigger, or creating a function that called a procedure that executed **RENAME TABLE** could, under certain circumstances, raise an assertion. (Bug #74740, Bug #19988193, Bug #21198646)

• Timestamps for server-side prepared statements could be written to the binary log up to a second behind timestamps for the corresponding nonprepared statements, leading to time value differences between master and slave servers. (Bug #74550, Bug #19894382, Bug #25187670)

• For dynamic storage engine plugins, **DROP TABLE**, **TRUNCATE TABLE**, and **RENAME TABLE** did not work due to incorrectly determining the engine from the **.frm** file. (Bug #74277, Bug #19902868)

• Executed prepared statements are logged with ? parameter markers replaced by data values. Construction of the logged string was inefficient and has been improved. (Bug #73056, Bug #20955496)

• Assignment by a plugin to its thread variables of string type could leak memory. (Bug #71759, Bug #19917521)

• Grouping with a view could produce an **ER_INVALID_GROUP_FUNC_USE** error (“Invalid use of group function”) when selecting from the base table did not. (Bug #70220, Bug #17406425)

• Test cases that were intended to be storage engine-agnostic but were actually using a specific engine were corrected.

## Index

### Symbols

- `include`, 90, 148
- `includedir`, 90, 148
- `--basedir`, 169
- `--binary-as-hex`, 50, 169
- `--column-statistics`, 169
- `--comments`, 169
- `--daemonize`, 169
- `--des-key-file`, 148
- `--gdb`, 74, 90
- `--get-server-public-key`, 114
- `--hex-blob`, 74
- `--ignore-built-in-innodb`, 148
- `--initialize`, 169
- `--log-warnings`, 148
- `--network-timeout`, 194
MySQL 8.0 Release Notes

--no-beep, 50
--no-dd-upgrade, 3
--no-monitor, 74
--secure-auth, 148
--server-public-key-path, 114
--skip-comments, 169
--skip-grant-tables, 50, 74, 148
--skip-log-bin, 114
--skip-name-resolve, 28
--skip-networking, 148
--skip-symbolic-links, 114
--ssl, 236
--ssl-cipher, 114
--ssl-fips-mode, 3, 50
--ssl-mode, 236
--ssl-mode=VERIFY_IDENTITY, 74
--ssl-verify-server-cert, 74, 236
--symbolic-links, 114, 169
--temp-pool, 194
--transaction-isolation, 148
--transaction-read_only, 148
--upgrade, 3
.mysql_secret, 3
1F, 148
32-bit, 236
\u001f, 148
, 74, 114

A
Aborted_connects, 3
account categories, 3
account management, 3, 50, 90, 169, 236
adaptive hash index, 114, 194, 236
ADD COLUMN, 28, 74
AddressSanitizer, 90
add_not_null_conds(), 3
ADMIN, 74
dm_admin_address, 28
admin_port, 28
AES_DECRYPT(), 114
aggregation, 114, 194
ALGORITHM, 3
ALGORITHM=COPY, 236
ALGORITHM=INSTANT, 74
alias, 3
ALLOW_INVALID_DATES, 3
Alpine Linux, 90, 114, 169
ALTER COLUMN, 50
ALTER RESOURCE GROUP, 74
ALTER TABLE, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
ALTER TABLESPACE, 28, 114
ALTER USER, 28, 90
ANALYZE TABLE, 90, 169
ANALYZE TABLE UPDATE HISTOGRAM, 50
ARCHIVE, 28
ARM 64-bit, 50
arrays, 114
ASC, 50, 74
ASCII 31, 148
ASSERT_COLUMN_MARKED_FOR_READ, 28
asynchronous I/O, 3
atomic DDL, 90, 148, 194, 236
atomic operations, 236
audit plugins, 3
audit_api_message_emit, 28
audit_api_message_emit_udf(), 28
audit_log plugin, 3, 28, 50, 74, 90, 114, 194
authentication plugins, 3, 28, 50, 90, 114, 148
auth_sock plugin, 90
auto-increment, 148, 169, 236
autocommit, 114
automatic_sp_privileges, 74
autopositioning, 169
autoscale, 148
autowrapping, 148
auto_increment_increment, 28
auto_increment_offset, 28
AVG(), 28, 50

B
B-tree, 50
backup, 148
back_log, 148
Barracuda, 236
base-64, 74
basedir, 169
BETWEEN, 50
big_tables, 114
BIN(), 194
binary, 50
binary log, 148, 169, 194, 236
bind_address, 50
binlog_direct_non_transactional_updates, 28
binlog_expire_logs_seconds, 74, 114, 194
binlog_format, 28, 169
binlog_rows_query_log_events, 28
binlog_row_image, 28
binlog_row_value_options, 28, 114, 148
binlog_transaction_dependency_history_size, 194
binlog_transaction_dependency_tracking, 114, 194
BIN_TO_UUID(), 236
BIT, 236
bit operations, 236
BIT_AND(), 74
BIT_OR(), 74
BIT_XOR(), 74
MySQL 8.0 Release Notes

BLOB, 3, 90, 114, 236
bool, 194
Boost, 3, 28, 74, 90, 114, 169, 194, 236
bootstrap, 114, 169
BUCKETS, 90
buffer pool, 50, 74, 90, 114, 148, 194, 236
buffer pool mutex, 236
buffer pool scan, 236
bulk_insert_buffer_size, 28
BUNDLE_RUNTIME_LIBRARIES, 90

C
C API, 3, 114, 148, 169, 194, 236
C++11, 236
C++14, 3
caching_sha2_password, 3, 90, 114, 148
caching_sha2_password_auto_generate_rsa_keys, 114
capture groups, 74
CAP_SYS_NICE, 28
CASE, 194
CAST, 74
CAST(), 236
CEIL(), 236
change buffer, 236
change buffering, 114
CHANGE MASTER TO, 114
CHANGE REPLICATION FILTER, 169
channel initialization, 169
CHAR, 28
CHAR(), 74
character escapes, 148
character sets, 3, 28, 50, 90, 114, 148, 169, 194, 236
character_set_connection, 114
character_set_database, 28, 194
character_set_filesystem, 28
character_set_server, 194
charsets, 50
CHECK, 3
CHECK constraints, 3
CHECK TABLE, 3, 28, 50
checkpoint, 50
checksum, 50, 236
checksum mismatch, 74
CHECKSUM TABLE, 236
check_stack_overrun(), 194
Chinese, 3
Clang, 169, 236
CLIENT_SSL_VERIFY_SERVER_CERT, 90
CMake, 3, 28, 50, 90
CMAKE_BUILD_TYPE, 28, 50
COALESCE, 3
COALESCE(), 114
COLLATE, 148
collations, 3, 50, 90, 148, 169, 194, 236
collation_database, 28, 194
collation_server, 194
columns_priv table, 236
column_statistics table, 148, 169
COLUMN_STATISTICS table, 169
column_stats table, 169
comments, 50, 169
common table expressions, 28, 114, 148, 194
COMPILATION_COMMENT, 28
COMPILATION_COMMENT_SERVER, 3, 28
COMPILATION_DEFINITIONS, 3
COMPILE_FLAGS, 3
compiling, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
components, 28, 90, 114, 169, 194, 236
component_sys_variable_register service, 169
component_sys_variable_unregister service, 169
composite indexes, 28
compression, 148, 194, 236
COM_SHUTDOWN, 236
CONCAT(), 3, 114
condition pushdown, 3
correlation, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
cost model, 194
cost tables, 194
COUNT(), 3, 28, 114, 169
COUNT(*), 50
const, 3, 74
column folding, 3
constant propagation, 114
constraints, 3
conversions, 28
CONVERT(), 50, 236
core file, 28
corruption, 236
cost model, 194
cost tables, 194
COUNT(), 3, 28, 114, 169
COUNT(*), 50
crc32, 236
CREATE FUNCTION, 169
CREATE INDEX, 74, 194
CREATE MASTER, 236
CREATE RESOURCE GROUP, 74
CREATE SPATIAL REFERENCE SYSTEM, 74, 114
CREATE TABLE, 3, 28, 74, 90, 114, 148, 169, 194, 236
CREATE TABLE ... SELECT, 90
CREATE TABLE LIKE, 90, 114
CREATE TABLE SELECT, 90
CREATE TABLESPACE, 28, 169
CREATE TEMPORARY TABLE, 114, 236
CREATE USER, 3, 28, 90, 148
CREATE VIEW, 90, 114
create_admin_listener_thread, 28
CREATE_OPTIONS, 114
cte_max_recursion_depth, 114, 148
CTX_NONE, 114
CTX_WINDOW, 114
curl, 3, 50
CURRENT_ROLE(), 90
CURRENT_USER, 3
CURSOR_TYPE_READ_ONLY, 90

data dictionary, 3, 28, 28, 50, 74, 90, 114, 148, 169, 194, 236
data dictionary cache, 169
DATA DIRECTORY, 28
data file, 236
data masking, 50
data types, 50, 148, 236
data_locks table, 28, 50
data_lockwaits table, 50
DATE, 3, 90
date arithmetic, 114
DATETIME, 3, 90
datetime_format, 148
date_format, 148
db table, 236
DEBUG_ABORT(), 50, 169
DDL, 3, 50, 74, 90, 114, 148, 194
ddl_rewriter, 3
dd_properties table, 169
deadlock, 28, 74, 90, 236
deadlock detection, 28
debug, 28, 50, 74, 90, 114, 236
Debug Sync, 194
ddebug_sync, 28
DECODER(), 148
DEFAULT, 50
default values, 50
default_authentication_plugin, 114
default_collation_for_utf8mb4, 28, 90
default_table_encryption, 3
DELETE, 3, 28, 74, 90
DENSE_RANK(), 114
DENSE_RANK() OVER, 114
deprecation, 3
derived tables, 3, 28, 114, 194
DESC, 50, 74
descending indexes, 194
DESCRIBE, 90
DES_DECRYPT(), 148
DES_ENCRYPT(), 148
dictionary object cache, 236
disabled_storage_engines, 74
DISCARD OLD PASSWORD, 28
DISCARD TABLESPACE, 74
disk is full, 114
disk space, 114
DISTINCT, 3, 74, 169
DML, 236
Docker, 3, 28, 148
DOM, 194
double, 194
downgrade, 90
doxygen, 90, 236
dragnet.log_error_filter_rules, 74, 114
dragnet.Status, 74
DROP DATABASE, 74, 114, 148, 194
DROP DATAFILE, 169
DROP FUNCTION, 169
DROP INDEX, 194
DROP RESOURCE GROUP, 74
DROP ROLE, 3
drop role, 3
DROP SPATIAL REFERENCE SYSTEM, 114
DROP TABLE, 74, 148, 169, 194
DROP TABLESPACE, 169
DROP VIEW, 148
DTrace, 194
dual passwords, 28
duplicate keys, 148
dynamic metadata, 90

E
embedded server, 194
empty string, 236
ENCODE(), 148
ENCRYPT(), 148
encryption, 3, 28, 74, 90, 114, 148, 169, 194
end-range condition, 114
ENUM, 28, 169
EPSG, 74, 114
equi-joins, 114
eq_ref, 3
error log, 90
error logging, 114, 169
errors, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
ER_IB_MSG_720, 28
ER_INCONSISTENT_ERROR, 169
ER_NEED_REPREPARE, 28
ER_REGEXP_ILLEGAL_ARGUMENT, 3
ER_REGEXP_INVALID_CAPTURE_GROUP_NAME, 90
ER_WRONG_FIELD_WITH_GROUP, 3
ER_WRONG_VALUE_FOR_VAR, 28
Event Scheduler, 169
events table, 236
events_statements_current, 28
events_statements_history, 28
events_statements_history_long, 28
events_statements_summary_by_digest, 148
event_scheduler, 148
EXCHANGE, 90
EXCHANGE PARTITION, 3, 28
EXISTS, 3
date_add, 74, 114
expire_logs_days, 74, 114
EXPLAIN, 3, 50, 74
EXPLAIN EXTENDED, 90, 148
EXPLAIN FOR CONNECTION, 28
EXPLAIN PARTITIONS, 148
explicit_defaults_for_timestamp, 28, 114, 169
exponents, 50
evaluation default values, 50
expressions, 3
EXTENDED, 169
extent, 236
false, 3
FEDERATED, 28, 50, 74, 194
Field_json::cmp_binary(), 114
FILES, 90, 114
filesort, 28, 74
FIND_IN_SET(), 50
FIPS, 90
firewall, 28, 74
FIRST_VALUE(), 90
FLOAT, 50
floating-point numbers, 114
FLOOR(), 236
flush, 114
FLUSH DES_KEY_FILE, 148
FLUSH LOGS, 169
flush method, 236
FLUSH QUERY CACHE, 148
FLUSH STATUS, 3
FLUSH TABLES, 114, 148, 169
FLUSH TABLES FOR EXPORT, 50
flushing, 148
FOLLOWING, 28
FORCE INDEX, 148
FORCE_INSOURCE_BUILD, 28
foreign key check, 50
foreign key constraint, 3, 50, 148, 194
foreign keys, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
formatting, 169
FORMAT_BYTES(), 3
FORMAT_PICO_TIME(), 3
fragment page, 236
FreeBSD, 194
frm files, 50
FROM_UNIXTIME(), 3, 114
fsync, 50
full-text, 90, 236
full-text search, 28, 74, 90, 114, 148, 169, 194
FULLTEXT index, 50, 74, 169, 194
func table, 236
functional dependencies, 90
functional indexes, 28, 50
functional key parts, 50
functions, 3, 194
futex, 194

gap lock, 194
gap locks, 90
general tablespace, 148
generated columns, 3, 28, 50, 74, 90, 114, 169, 194, 236
GeomCollection, 90
GeomCollection(), 90
geometry, 194
GeometryCollection, 90
GeometryCollection(), 90
GET_LOCK(), 28, 114
get_mysql_time_from_str_no_warn(), 3
GIS, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
glibc, 194
global_variables, 114
GRANT, 3, 90, 114
GREATEST(), 90, 114, 236
group, 3, 28, 50, 74, 90, 114, 148, 169, 194
Group, 114, 148
GROUP BY, 3, 50, 74, 90, 114
group commit, 194, 236
Group Replication, 50
group replication, 169
grouping, 74, 114
GROUPING(), 194
GROUPING(), 114, 169, 194
group_concat_max_len, 114
group_replication_recovery_get_public_key, 114
group_replication_recovery_public_key_path, 114
GTID, 169, 194, 236
GTID skip, 169
gtid_executed, 74, 114
gtid_mode, 148, 169
gtid_next, 28

Handler::rnd_pos(), 90
hash semi-join, 114
have_crypt, 148
have_query_cache, 148
have_symlink, 169
HAVING, 114
heap size, 169
heap-use-after-free, 90
Henry Spencer Library, 3
HEX(), 236
hexadecimal, 74
HISTOGRAM, 114
histogram, 114
histogram statistics, 148, 169
histograms, 3, 50
histogram_generation_max_mem_size, 28, 169
history list, 114
host_application_signal component service, 28

ibd2sdi, 74, 90, 148, 169, 236
ibtmp1, 236
ICU, 114
ICU_VERSION(), 114
IF(), 114
IFNULL(), 114
IGNORE, 28
ignore_builtin_innodb, 148
IGNORE_SPACE, 169
implicit tablespace, 114
import, 114, 236
IMPORT TABLE, 194
Important Change, 28, 50, 74, 90, 114, 194, 236
Important Note, 50
IN, 28, 114
IN subqueries, 3
in-place update, 169
Incident event, 169
Incompatible Change, 3, 50, 74, 90, 114, 148, 169, 194, 236
index, 194
index condition pushdown, 114
index dives, 148
index statistics, 236
Index.xml, 90
indexes, 50, 74, 90
index_id, 236
INDEX_MERGE, 194
Information Schema, 114
INFORMATION_SCHEMA, 28, 50, 74, 90, 114, 148, 169, 194, 236
INFORMATION_SCHEMA.COLUMNS, 148
INFORMATION_SCHEMA.COLUMN_STATISTICS, 114
INFORMATION_SCHEMA.FILES, 194
INFORMATION_SCHEMA.INNODB_COLUMNS, 74
INFORMATION_SCHEMA.INNODB_SYS_FOREIGN, 148
INFORMATION_SCHEMA.KEYWORDS, 50
INFORMATION_SCHEMA.STATISTICS, 90
INFORMATION_SCHEMA.TABLES, 3, 114
initialize, 50
init_connect, 90
innochecksum, 3, 114, 236
InnoDB, 3, 28, 28, 50, 74, 90, 114, 148, 169, 194, 236
INNODB_CACHED_INDEXES, 236
innodb_ddl_log table, 90
innodb_dedicated_server, 28
innodb_dynamic_metadata table, 90
innodb_file_format, 236
innodb_file_format_check, 236
innodb_file_format_max, 236
innodb_flush_log_at_trx_commit, 50
innodb_flush_method, 3, 28
INNODB_FOREIGN, 3
innodb_fsync_threshold, 50
innodb_large_prefix, 236
innodb_log_checkpoint_fuzzy_now, 74
INNODB_METRICS, 3
innodb_numa_interleave, 194
innodb_spin_wait_delay, 28
innodb_spin_wait_pause_multiplier, 3
INNODB_TRX table, 50
INSERT, 74, 114, 194
INSERT ... ON DUPLICATE KEY UPDATE, 3, 90
INSERT IGNORE, 114
INSTALL COMPONENT, 90, 114, 236
installing, 3, 28, 50, 74, 90, 114, 169, 194
INSTALL_STATIC_LIBRARIES, 236
internal SQL parser, 236
internal temporary table, 194
internal_tmp_disk_storage_engine, 3
introducers, 114
IS NOT NULL, 3
isl file, 236
IS_UUID(), 236
Item_subselect::walk_body(), 3

J
Japanese, 169, 194
joins, 3
JSON, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
json, 114
JSON functions, 28
Json_array, 148, 169
JSON_ARRAYAGG(), 28, 50, 194
JSON_ARRAY_INSERT(), 148
JSON_CONTAINS(), 236
JSON_CONTAINS_PATH(), 194
Json_dom, 148
JSON_EXTRACT(), 148, 236
JSON_INSERT(), 114, 148
JSON_MERGE(), 148, 236
JSON_MERGE_PATCH(), 148
JSON_MERGE_PRESERVE(), 148
Json_object, 148
JSON_OBJECT(), 114
Json_object::consume(), 236
JSON_OBJECTAGG(), 28, 50, 194
Json_path, 90
JSONPRETTY(), 194
JSONQUOTE(), 90, 194
JSON_REMOVE(), 169
JSON_REPLACE(), 169
JSON_SEARCH, 194
JSON_SEARCH(), 90, 236
JSON_SET(), 50, 114, 169, 236
JSON_STORAGE_FREE(), 169
JSON_STORAGE_SIZE(), 169
JSON_TABLE(), 28, 50, 74, 90, 114
JSON_TYPE(), 236
JSON_UNQUOTE, 194
JSON_UNQUOTE(), 50, 194, 236
JT_CONST, 114

K
kana sensitivity, 169
keyring, 3, 28, 74, 90, 114
keyring migration, 114
keyring_encrypted_file plugin, 28, 74, 114
keyring_file plugin, 90, 194
keyring_keys, 3
keyring_okv plugin, 28, 74, 194
keyring_operations, 74
keywords, 3
KEYWORDS, 90
KEY_BLOCK_SIZE, 169
KEY_COLUMN_USAGE, 28
KILL, 3, 90

L
LAG(), 114
large object data, 74, 114
last (keyword), 169
last_committed, 169
LAST_INSERT_ID(), 3
Last_query_cost, 3
LAST_VALUE(), 90, 114
latch, 169
latch debugging, 194
LATERAL, 3, 28
lateral derived tables, 28
latin1, 169, 194
latin1_swedish_ci, 194
lc_messages, 169
lc_messages_dir, 169
LDAP, 3, 28, 50, 90, 114
LDML, 90
LEAD(), 114
LEAST(), 90, 114, 236
LEFT JOIN, 90
left outer join, 114
LEFT(), 236
libevent, 50, 236
libmysqlclient version, 236
libmysqld, 194, 236
libtirpc, 90
LIBWRAP, 28
LIKE, 28, 90
LIKE ... ESCAPE, 3
LimitNOFILE, 90
LINK_RANDOMIZE, 74
Linux, 3, 90, 114, 169
LIST, 148
literal values, 236
LOAD DATA, 114, 169, 194
LOAD XML, 169
LOAD_DATA(), 3
LOB, 3, 50, 74, 90, 194
LOB pages, 28
local_infile, 169
LOCATE(), 194
LOCK INSTANCE FOR BACKUP, 90
LOCK TABLES, 50, 114
locking, 3, 28, 50, 90, 114, 148, 169, 194, 236
locking read, 194
locks, 3, 28, 50, 169
lock_wait_timeout, 50
log buffer, 74, 236
log rotation, 194
log-error, 28
logging, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
log_builtin_as_identified_by_password, 90
log_error_services, 148, 169
log_error_suppression_list, 50
log_error_verbosity, 114, 148
log_filter_dragnet, 114
log_filter_internal, 114
log_sink_internal, 114
log_sink_syseventlog, 50
log_slow_extra, 28
log_status, 90
log_syslog, 50
log_syslog_facility, 50
log_syslog_include_pid, 50
log_syslog_tag, 50
log_warnings, 148
Loose Index Scan, 114
lowercase_table_names, 28, 50
lower_case_table_names, 28, 90
LPAD(), 50
lz4_decompress, 236
MySQL 8.0 Release Notes

M
macOS, 3, 50, 74, 90, 114
materialized tables, 114
MAX(), 114
max_allowed_packet, 148
max_connections, 148
max_error_count, 148
max_execution_time, 50
max_length_for_sort_data, 194
max_tmp_tables, 148
mbind, 50
MBRContains(), 148, 169
MBRCoveredBy(), 148, 169
MBRCovers(), 148, 169
MBREquals(), 148, 169
MBRIntersects(), 148, 169
MBROverlaps(), 114, 148, 169
MBRTouches(), 148, 169
MBRWithin(), 148, 169
MeCab, 74, 114
memcached, 50, 74, 114, 148, 169, 194, 236
MEMORY, 50, 114
memory, 169, 194
memory allocation, 114
MERGE, 50, 236
metadata, 28, 169, 236
metadata locking, 114
metadata_locks_cache_size, 50
metadata_locks_hash_instances, 50
Microsoft Windows, 3, 28, 50, 74, 90, 114, 169, 194, 236
MSVC, 3
MTR_UNIQUE_IDS_DIR, 3
MTS, 114, 148, 169, 194
multi-source, 169
multithreaded slave, 148, 169, 194
multi_range_count, 148
mutex, 28, 74, 236
MVCC, 28, 50, 148
MyISAM, 74
myisam_use_mmap, 114
mysql-test-run.pl, 3, 169, 194, 236
mysql-test.run.pl, 194
mysql.gtid_executed, 114, 169
mysql.innodb_table_stats.table_name, 236
mysql.session, 74
mysql.session account, 169
mysqladmin, 3
mysqlbinlog, 28, 148, 169
mysqld, 50
mysqld-auto.cnf, 50, 90
mysqldump, 28, 50, 74, 114, 148, 169, 194, 236
mysqld_multi, 28
mysqld_safe, 28, 194
mysqld_safe.pid, 194
mysqlpump, 3, 28, 74, 90, 114, 148, 169, 194
mysqltest, 114, 169, 194, 236
mysqltest_safe_process, 3
mysqlxtest, 114, 169
mysqlxtext, 74
mysqlx_interactive_timeout, 114
mysqlx_read_timeout, 114
mysqlx_wait_timeout, 114
mysqlx_write_timeout, 114
mysql_config, 90, 194
mysql_config_editor, 194
mysql_fetch_row_nonblocking(), 3
mysql_free_result_nonblocking(), 3
MYSQL_GROUP_SUFFIX, 114
MYSQL_HOME, 3
mysql_insert_id(), 194
mysql_install_db, 3, 236
mysql_install_plugin, 50
mysql_native_password, 114
mysql_next_result_nonblocking(), 3
mysql_options(), 169
MYSQL_OPT_SSL_MODE, 169
mysql_plugin, 236
mysql_real_connect_nonblocking(), 3
mysql_real_query_nonblocking(), 3
mysql_reset_server_public_key(), 114
mysql_result_metadata(), 50
mysql_secure_installation, 3, 74, 194
MYSQL_SERVER_PUBLIC_KEY, 90
mysql_service_udf_registration service, 169
mysql_service_udf_registration_aggregate service, 169
mysql_shutdown(), 194, 236
mysql_ssl_rsa_setup, 3, 114
mysql_stmt_close(), 169
mysql_stmt_errno(), 169
mysql_stmt_error(), 169
mysql_stmt_sqlstate(), 169
mysql_store_result_nonblocking(), 3
mysql_string component, 169
mysql_upgrade, 3, 90, 114, 169
mysys, 236
my_bool, 194
my_init(), 169

N

named_pipe_full_access_group, 3, 28
NDB Client Programs, 50
NDB Cluster, 3, 50, 236
ndb_binlog_index, 236
ndb_cache_check_time, 148
ngram, 74, 148
NO PAD collations, 28, 169, 194
NOOP, 114
NOT NULL, 90, 114
NOTIFY_SOCKET, 90
NOWAIT, 194
NO_AUTO_CREATE_USER, 50, 90
NO_ENGINE_SUBSTITUTION, 74
NO_INDEX_MERGE, 194
NO_MERGE, 236
NO_ZERO_DATE, 90
NTH_VALUE(), 114
NULL, 3, 74, 90, 114, 148, 194, 236
NULLIF(), 50, 169
NUMA, 50, 169, 236

O
offline_mode, 28
OFFSET, 3
old_passwords, 90, 114
ON_DUPLICATE_KEY_UPDATE, 114, 194
one_or_all, 194
online DDL, 28, 74, 90, 114
only_full_group_by, 90
ONLY_FULL_GROUP_BY, 194
OpenSSL, 3, 28, 50, 74, 90, 114, 169, 194
operators, 28, 236
optimization, 3
OPTIMIZE PARTITION, 169
OPTIMIZE TABLE, 50
optimizer, 28, 50, 74, 90, 114, 148, 169, 194, 236
optimizer hints, 148, 194, 236
optimizer_switch, 148
optimizer_trace_max_mem_size, 114
optimizer_trace_offset, 90
options, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
ORDER BY, 3, 28, 74, 114, 194
original_commit_timestamp, 28, 169
outer joins, 3, 28

P
packaging, 3, 28, 28, 50, 74, 90, 114, 148, 169, 194, 236
Packaging, 74, 114, 194
PAD SPACE collations, 194
PAD_CHAR_TO_FULL_LENGTH, 50
page, 236
page cleaner, 236
page cleaner thread, 194
page compression, 90, 148, 169
page tracking, 3
parallel index read, 28
parallelization, 114, 194
parser, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
parser_max_mem_size, 50
partial indexes, 90
partial revokes, 3
partial update, 114, 169
partial updates, 114, 148
partition, 148, 236
partition ID, 148
Partitioning, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
partitioning, 28, 74, 114, 148, 169, 194
password expiration, 114
password history, 148
password verification, 50
PASSWORD(), 3, 90, 114
passwords, 28, 50, 148
password_history, 148
password_require_current, 50
password_reuse_interval, 148
PATH, 50
path expressions, 148
paths, 236
perdicate elimination, 28
performance, 74, 236
Performance, 148, 169
Performance Schema, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
PERFORMANCE_SCHEMA.DATA_LOCKS, 3
performance_schema_max_mutex_classes, 148
PERIOD_ADD(), 3, 114
PERIOD_DIFF(), 114
perror, 90
PERSIST, 3
persisted variables, 90
PERSIST_ONLY, 3
persist_only_admin_x509_subject, 28
PERSIST_RO_VARIABLES_ADMIN, 74, 148, 169
pid_file, 28
pkg-config, 194
pluggable authentication, 3, 28, 90, 114, 148, 169
plugin, 169
plugin service, 148
plugins, 3, 28, 50, 74, 90, 114, 148, 169, 194
plugin_registry_service, 194
pointers, 148
prefixes, 50, 90
PREPARE, 74
prepared statements, 3, 28, 74, 90, 114, 169, 236
primary key, 50
primary passwords, 28
privileges, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
proc table, 236
procedure, 236
PROCEDURE ANALYSE(), 194
proxies_priv table, 236
PROXY, 169
pruning, 3, 236
pseudo_slave_mode, 28
pseudo_thread_id, 28
PS_CURRENT_THREAD_ID(), 3
PS_THREAD_ID(), 3
purge, 3, 28, 236
pushdown, 3

Q
Qcache_free_blocks, 148
Qcache_free_memory, 148
Qcache_hits, 148
Qcache_inserts, 148
Qcache_lowmem_prunes, 148
Qcache_not_cached, 148
Qcache_queries_in_cache, 148
Qcache_total_blocks, 148
query cache, 148, 169, 194
query cost, 3, 90
query rewrite plugins, 3, 74, 169
query_cache_limit, 148
query_cache_min_res_unit, 148
query_cache_size, 148, 194
query_cache_type, 148
query_cache_wlock_invalidate, 148
QUICK_GROUP_MIN_MAX_SELECT, 114
QUOTE(), 3, 114, 194
quoting, 148

R
R-tree, 90, 194
race condition, 236
range, 90
range access, 114
range frames, 3, 28
range_index access, 28
ranges, 169
range_optimizer_max_mem_size, 50
RapidJSON, 50, 194
rapidjson, 194
rbr_exec_mode, 28
read-ahead, 194
read_only, 169
receiver thread, 169
recovery, 3, 74, 90, 114, 148, 169, 194, 236
redo, 28, 50
redo log, 3, 28, 50, 90, 114, 169, 194
ref access, 28, 114
regex, 3
regexp, 90
REGEXP, 114
regexp functions, 3, 28
regexp::EvalExprToCharset(), 50
REGEXP_INSTR(), 74, 114
REGEXP_LIKE(), 74, 114
REGEXP_REPLACE(), 74, 114
regexp_stack_limit, 114
REGEXP_SUBSTR(), 74, 114
regexp_time_limit, 114
register, 236
regular accounts, 3
regular expressions, 90, 114
regular sessions, 3
relay log, 169, 194
RENAME COLUMN, 148
RENAME TABLE, 3, 28, 50, 90, 169, 194
RENAME USER, 74, 90
REPEAT(), 236
REPEATABLE-READ, 169
REPLACE, 74, 90, 114
REPLACE(), 50, 148, 194, 236
replace_numeric_round, 169
Replication, 3, 28, 50, 74, 90, 114, 148, 169, 194, 236
replication, 114, 148, 236
replication_applier_status_by_worker, 148, 169
replication_connection_configuration, 114
REPRODUCIBLE_BUILD, 74
reserved user accounts, 114, 169
reserved words, 194, 236
RESET PERSIST, 28, 74, 90, 194
RESET QUERY CACHE, 148
resolveip, 28
resolve_stack_dump, 28
resource groups, 28, 148
resource usage, 236
RESTART, 74, 90, 114
restart, 169
RETAINT CURRENT PASSWORD, 28
REVOKE, 114
RFC 7159, 148
RIGHT(), 236
RLIKE, 114
roles, 3, 50, 74, 90, 114, 148, 169, 194, 236
ROLES_GRAPHML(), 90
rollback, 3, 50, 114, 236
rollback segment, 194
rollback segments, 169
ROLLUP, 3, 28, 50, 90, 114, 194
Rotate_log_event, 169
ROW, 90
ROW FORMAT, 169
row format, 236
row-based, 194
row-based replication, 114
ROW_FORMAT, 114, 194
row_table_add_foreign_constraints(), 148
RPAD(), 50
RPC, 28, 90
rpcgen, 90
RPM, 169
Russian, 148

S
SAFE_MUTEX, 148
savepoints, 50
scope, 148
secondary passwords, 28
secure_auth, 148
secure_file_priv, 194
security, 3, 28, 50, 169
SELECT, 28, 50, 90, 194
select_lex, 50
semi-join, 114
semi-joins, 3
serialization, 114
serialized dictionary information (SDI), 3, 50, 90, 114, 148, 169
server initialization, 90
session categories, 3
SESSION_VARIABLES_ADMIN, 28
SET, 169
SET DEFAULT ROLE, 114
SET GLOBAL, 169
SET PASSWORD, 3, 28, 90
SET PERSIST, 28, 90, 114, 148, 169, 236
SET PERSIST_ONLY, 3, 28, 74, 90, 114, 169
SET RESOURCE GROUP, 28
SET ROLE, 3, 74
SET statement, 50
setuid, 74
setup_instruments, 148
setup_threads, 148
SET_VAR, 114, 148
sha256 cache, 114
sha256_password, 3, 114, 148
sha256_password_auto_generate_rsa_keys, 114
SHOW COLUMNS, 148, 169
SHOW CREATE TABLE, 28, 50, 74, 90, 114, 169, 236
SHOW CREATE TRIGGER, 3
SHOW CREATE USER, 74, 148
SHOW CREATE VIEW, 148
SHOW ENGINE INNODB STATUS, 236
SHOW GRANTS, 74, 90
SHOW INDEX, 169
SHOW INDEXES, 50
SHOW PLUGINS, 114
SHOW PROCESSLIST, 90
SHOW RELAYLOG EVENTS, 169
SHOW SLAVE STATUS, 90, 114, 169
SHOW TABLE STATUS, 74, 114
SHOW TABLES, 169
show_create_table_verbosity, 90
shutdown, 74, 90, 169
SHUTDOWN, 236
SIGHUP, 28
signals, 28, 74
SKIP LOCKED, 194
Skip Scan, 50
skip-slave-start, 169
skip_name_resolve, 3
slave, 236
slave_applier, 169
slave_pending_jobs_size_max, 74, 169
slave_skip_errors, 169
slow query log, 28
slow shutdown, 236
Solaris, 28, 50, 90, 169, 194, 236
sort, 194
sort buffer, 3
sort keys, 50
space_id, 236
SPATIAL, 74
spatial index, 50, 114, 148
spin lock polling, 3
SQL, 169
SQL syntax, 236
sql_auto_is_null, 3
sql_buffer_result, 148
SQL_CACHE, 148, 236
sql_log_bin, 28, 148
sql_log_off, 28
sql_mode, 28, 50, 74, 90, 169, 194
SQL_NO_CACHE, 114, 148, 236
sql_require_primary_key, 3, 28, 50
sql_safe_updates, 50
SRID attribute, 148
Srv_session, 3
SSL, 3, 28, 50, 74, 90, 114, 169, 194
ssl_fips_mode, 50
stack size, 194
standard monitor, 90
startup, 3, 50, 74, 90, 114, 169, 236
startup configuration, 90
STATEMENT_DIGEST(), 114
STATEMENT_DIGEST_TEXT(), 114
statistics, 194, 236
status_variable_registration service, 169
std::vector, 169
storage engines, 236
stored procedures, 3, 50, 74, 114
stored programs, 28, 74, 90, 114, 148, 169, 236
stored routines, 114
STRAIGHT_JOIN, 236
strict mode, 3
STRING, 3
storage engines, 236
strings, 3
ST_Area(), 50, 114
ST_AsBinary(), 194
ST_AsGeoJSON(), 114, 236
ST_AsText(), 114, 194
ST_AsWKB(), 194
ST_AsWKT(), 194
ST_Buffer(), 114
ST_Centroid(), 114
ST_Contains(), 148, 169
ST_Centroid(), 114
ST_Crosses(), 114, 148, 169
ST_Difference(), 114
ST_Dimension(), 114
ST_Disjoint(), 148, 169
ST_Distance(), 28
ST_Distance_Sphere(), 114
ST_EndPoint(), 114
ST_Envelope(), 114
ST_Equals(), 148, 169
ST_ExteriorRing(), 114
ST_Geohash(), 114
ST_GeomCollFromText(), 114, 194
ST_GeomCollectionFromText(), 114, 194
ST_GeomCollFromWKB(), 114, 194
ST_GeomCollectionFromWKB(), 114, 194
ST_GeometryCollectionFromText(), 114, 194
ST_GeometryCollectionFromWKB(), 114, 194
ST_GeometryFromText(), 114, 194
ST_GeometryFromWKB(), 114, 194
ST_GeometryN(), 114
ST_GeometryType(), 90, 114
ST_GeomFromGeoJSON(), 114, 236
ST_GeomFromText(), 114, 194
ST_GeomFromWKB(), 114, 194
ST_InteriorRingN(), 114
ST_Intersection(), 114
ST_Intersection(), 114
ST_Equals(), 148, 169
ST_IsClosed(), 114
ST_IsEmpty(), 114
ST_IsValid(), 90, 114
ST_Latitude(), 74
ST_Length(), 3, 114
ST_LineFromText(), 114, 194
ST_LineFromWKB(), 114, 194
ST_LineStringFromText(), 114
ST_LinestringFromText(), 194
ST_LinestringFromWKB(), 114
ST_LinestringFromWKB(), 194
ST_Latitude(), 74
ST_MakeEnvelope(), 114
ST_MLineFromText(), 114, 194
ST_MLineFromWKB(), 114, 194
ST_MPointFromText(), 114, 194
ST_MPointFromWKB(), 114, 194
ST_MPolyFromText(), 114, 194
ST_MPolyFromWKB(), 114, 194
ST_MultiLineStringFromText(), 114
ST_MultiLineStringFromWKB(), 114
ST_MultiLineStringFromWKB(), 194
ST_MultiPointFromText(), 114, 194
ST_MultiPointFromWKB(), 114, 194
ST_MultiPolygonFromText(), 114, 194
ST_MultiPolygonFromWKB(), 114, 194
ST_NumGeometries(), 114
ST_NumInteriorRing(), 114
ST_NumInteriorRings(), 114
ST_NumPoints(), 114
ST_Overlaps(), 148, 169
ST_PointFromGeoHash(), 114
ST_PointFromText(), 114, 194
ST_PointFromWKB(), 114, 194
ST_PointN(), 114
ST_PolyFromText(), 114, 194
ST_PolyFromWKB(), 114, 194
ST_PolygonFromText(), 114, 194
ST_PolygonFromWKB(), 114, 194
ST_Simplify(), 50, 114
ST_SPATIAL_REFERENCE_SYSTEMS, 114
ST_SRID(), 236
ST_StartPoint(), 114
ST_SwapXY(), 114, 194
ST_SymDifference(), 114
ST_Touches(), 148, 169
ST_Transform(), 50
ST_Union(), 114
ST_UNITS_OF_MEASURE, 3, 28
ST.Validate(), 50, 114
ST_Within(), 148, 169
ST_X(), 74, 114, 236
ST_Y(), 74, 114, 236
subqueries, 3, 28, 50, 90, 114
SUBSTR(), 236
SUBSTRING_INDEX(), 236
SUM(), 50, 194
SUPER, 28, 74, 148
sync_binlog, 50
sys schema, 3, 90
sys.format_bytes(), 3
sys.format_time(), 3
sys.ps_thread_id(), 3
syseventlog.facility, 50
syseventlog.include_pid, 50
syseventlog.tag, 50
system accounts, 3
system database, 3
system sessions, 3
system variables, 28
systemd, 90
SYSTEM_VARIABLES_ADMIN, 28, 74, 148, 169

T

table encryption, 3
table functions, 114
table locks, 74
table statistics, 114, 148
TABLESPACE, 114
tablespace, 114, 148, 194, 236
tablespace encryption, 50, 74, 169, 194
tablespace import, 90
tablespace, 114, 169
tables_priv table, 236
table_definition_cache, 114
TABLE_ENCRYPTION_ADMIN, 3
table_encryption_privilege_check, 3
table_open_cache, 114
temporal data types, 114
temporary directory, 114
temporary tables, 50, 90, 114, 169, 194, 236
temporary tablespace, 50
TempTable, 3, 90, 148
TempTable storage engine, 3, 28, 50, 90, 148
temptable::Allocator, 28
test suite, 148, 169, 194, 236
thread management, 236
thread pool plugin, 28, 90
thread switching, 3
thread_cache_size, 3
thread_mutex, 50
thread_pool, 3
thread_stack, 74
TIME, 74
TIMESTAMP, 114, 169
time_format, 148
TIME_TRUNCATE_FRACTIONAL, 194
TLSv1.3, 3
to (keyword), 169
TO_SECONDS(), 3
transaction, 148
transaction locks, 148
transactions, 50, 114, 148, 169, 194
transaction_isolation, 148
transaction_read_only, 148
transaction_write_set_extraction, 28
transportable tablespace, 74, 169, 236
TREE, 3
triggers, 74, 114, 169, 194
trivial conditions, 28
TRUNCATE, 74
TRUNCATE PARTITION, 74, 236
TRUNCATE TABLE, 74, 90, 114, 169, 194, 236
truncation, 194
tuning, 148
two-phase commit, 236
tx_isolation, 148
tx_read_only, 148
type conversion, 3, 114, 236

U
UBSAN, 50, 90, 194
UBSan, 169
UDFs, 3, 148
udf_example, 114
UMASK, 114
UMASK_DIR, 114
Undefined Behavior Sanitizer, 50, 90
undo, 169
undo log, 3, 28, 50, 114, 194
undo log encryption, 3
undo tablespace, 3, 28, 50, 90, 114, 148, 194, 236
Unicode, 3, 148, 169, 194, 236
UNINSTALL COMPONENT, 90, 114, 236
UNION, 28, 50, 74, 90, 114, 194, 236
UNIQUE, 90
Unique::io_cache, 50
UNIV_INNOCHECKSUM, 236
UNIX_TIMESTAMP(), 114
UPDATE, 28, 74, 114, 194
upgrade, 3, 50, 74, 90, 114
upgrades, 3, 28, 114, 148, 169, 194
usability, 194
user table, 236
user variables, 3, 50
user-defined functions, 90
user_defined_functions table, 169
user_variables, 114
use_invisible_indexes, 148
USE_LD_LLD, 3
UTF-16, 194
utf32, 28, 90
utf8, 50, 90, 114
utf8mb3, 50, 90
utf8mb4, 3, 50, 90, 148, 169, 194
utf8mb4_0900_ai_ci, 194, 236
utf8mb4_0900_as_ci, 169
utf8mb4_general_ci, 194
utf8mb4_ja_0900_as_cs_ks, 169
utf8mb4_ru_0900_ai_ci, 148
utf8mb4_ru_0900_as_cs, 148
utf8mb4_zh_0900_as_cs, 3
UUID(), 148
UUIDs, 236
UUID_TO_BIN(), 236

V
Valgrind, 148, 236
validate_password, 28, 74, 90, 114
validate_password plugin, 236
variables, 28
variables_by_thread, 28
variables_info, 28, 90, 114, 148
VARIANCE(), 90
Variance-Aware Transaction Scheduling (VATS), 3, 50, 74, 90, 114, 148
VAR_POP(), 90
VAR_SAMP(), 90
VATS, 50
version table, 169
version_compile_zlib, 90
version_tokens plugin, 194
view references, 3
views, 28, 114, 194, 236
VIEW_ROUTINE_USAGE, 50
VIEW_TABLE_USAGE, 50
virtual column, 3
virtual columns, 28, 74, 90, 114, 148, 169, 194
virtual generated column, 236
virtual indexes, 3, 90, 194, 236
Visual C++ Redistributable, 194

W
warnings, 50
WEIGHT_STRING(), 114, 194
whitespace, 169
widen_fraction, 3
wildcards, 148
window functions, 3, 28, 50, 74, 90, 114, 148, 169
windowing functions, 3, 28, 114
windowing_use_high_precision, 90
Windows, 28
WITH, 194
WITH ROLLUP, 74
WITH_EDITLINE, 90
WITH_GMOCK, 50
WITH_ICU, 90, 114
WITH_INNODB_EXTRA_DEBUG, 90
WITH_JEMALLOC, 3
WITH_LIBEVENT, 90
WITH_LSAN, 3
WITH_LTO, 50
WITH_LZ4, 90
WITH_LZMA, 3, 114
WITH_MYSQLX, 3
WITH_PROTOBUF, 90
WITH_RAPIDJSON, 50
WITH_RE2, 114
WITH_ROUTER, 3
WITH_SSL, 90, 114
WITH_SYSTEM_LIBS, 90
WITH_ZLIB, 90
wolfSSL, 90

X
X DevAPI, 28
X Plugin, 28, 90
X Protocol, 3, 28, 50, 74, 90, 114, 148, 194, 236
X509, 114
XA, 50, 90, 114, 169, 194, 236
XA RECOVER, 169
XA transactions, 50, 114, 169, 194, 236
XA_RECOVER_ADMIN, 169
XDR, 28
XML, 194
XPath syntax, 169

Y
yassl, 50, 74
yaSSL, 90, 114, 148, 194
YEAR, 28, 50
YEAR(), 28
Yum repository, 114
YYYYMMDD, 90

Z
zlib, 90, 148
zlib_decompress, 236